

Dynamic and Geometric Contributions to
Digital Image Processing

Habilitation Thesis

by Martin Welk

submitted to

Department of Mathematics

Saarland University

in November 2007

Revised July 2013, February 2016

Contents

Introduction	7
Thanks and Acknowledgements	9
Structure	10
Revision Note	13
I Preliminaries	15
1 Single- and Multi-Channel Images	17
1.1 Grey-Value Images	18
1.2 Colour Images	19
1.3 Vector Fields	19
1.4 Matrix-Valued Images	20
1.4.1 Diffusion Tensor Images	21
1.4.2 Tiled Grey-Value Visualisation	22
1.4.3 Ellipsoid Visualisation	23
2 Diffusion Filter Models for Single- and Multi-Channel Images	29
2.1 Linear Diffusion	29
2.1.1 Basic Model	30
2.1.2 Variational Interpretation	30
2.1.3 Multi-Channel Linear Diffusion	32

2.2	Isotropic Nonlinear Diffusion	32
2.2.1	Basic Model	32
2.2.2	Variational Interpretation	34
2.2.3	Role of Pre-Smoothing	35
2.2.4	Special Singular Diffusivities	36
2.2.5	Multi-Channel Isotropic Nonlinear Diffusion	37
2.3	Anisotropic Nonlinear Diffusion	38
2.3.1	Basic Model	38
2.3.2	Multi-Channel Anisotropic Nonlinear Diffusion	40
2.3.3	Representation as Gradient Descent	40
II	Multichannel M-Smothers	43
3	M-Smothers and Related Filters for Multi-Channel Images	45
3.1	Median	47
3.1.1	Scalar-Valued Median	47
3.1.2	Multi-Channel Median Definition	48
3.1.3	Choice of Norms	50
3.1.4	Frobenius Median	52
3.1.5	Nuclear Median	55
3.1.6	Spectral Median	55
3.1.7	Numerics: Gradient Descent with Step-size Adaptation	57
3.1.8	Numerics: Convex Optimisation	59
3.1.9	Experimental Evaluation	60
3.2	Median-Related Filters	68
3.2.1	Weighted Median	68
3.2.2	Matrix-Valued Quantile Filtering	70

<i>Contents</i>	3
3.3 Midrange Filter	79
3.3.1 Scalar-Valued Midrange	79
3.3.2 Multi-Channel Midrange Definition	79
3.3.3 Properties of Multi-Channel Midrange Filters	79
3.3.4 Numerics	80
3.3.5 Experimental Evaluation	81
3.4 M-Estimators with Arbitrary Power Functions as Penalisers	83
3.4.1 Scalar-Valued M-Smothers and M-Estimators	83
3.4.2 Multi-Channel M-Smothers and M-Estimators	83
3.4.3 Numerical Aspects	85
3.5 Multi-Channel Median and Mean Curvature Motion	85
3.5.1 Multi-Channel Mean Curvature Motion	85
3.5.2 Multi-Channel Median and PDE Evolutions	87
III Dynamical Systems in Image Processing	97
4 Locally Analytic and Semi-Analytic Schemes for Diffusion Filters	99
4.1 Two-Pixel Approach to 1D Singular Differential Equations	102
4.1.1 Discretisation of 1D Total Variation Diffusion	102
4.1.2 Total Variation Diffusion on Two Pixels	103
4.1.3 A Numerical Scheme for 1D Total Variation Diffusion	107
4.1.4 Other Singular Nonlinear Diffusion Processes in 1D	108
4.1.5 Analytic Solution of 1D Total Variation Diffusion on n pixels	109
4.1.6 Haar Wavelet Interpretation	113
4.2 Four-Pixel Approach to Isotropic Nonlinear 2D Diffusion	117
4.2.1 Discretisation of the Divergence Form	117
4.2.2 Discretisation of Diffusivities	119

4.2.3	Variational Interpretation	120
4.2.4	Analytic Solutions	121
4.2.5	Locally Analytic Scheme for Singular Isotropic Diffusion Processes	125
4.2.6	Semi-Analytic Solution for Arbitrary Diffusivities	129
4.2.7	Locally Semi-Analytic Scheme for Isotropic Nonlinear Diffusion	131
4.2.8	Multi-Channel Images	132
4.2.9	Haar Wavelet Interpretation	132
4.3	Four-Pixel Approach to Anisotropic Nonlinear Diffusion	136
4.3.1	Discretisation of the Divergence Form	136
4.3.2	Discretisation of the Diffusion Tensor	139
4.3.3	Multi-Channel Case	140
4.3.4	Variational Interpretation in the Multi-Channel Case	140
4.3.5	Semi-Analytic Solution	142
4.3.6	Locally Semi-Analytic Schemes for Anisotropic Nonlinear Diffusion	145
4.3.7	Haar Wavelet Interpretation	148
4.3.8	Variational Interpretation with Fixed Diffusivity	149
4.4	Higher-Dimensional Extensions	151
4.4.1	Locally Analytic Scheme for Total Variation Diffusion in 3D	151
4.4.2	Anisotropic Diffusion in 3D	151
4.5	Splitting Schemes	157
4.5.1	Error Analysis	157
4.5.2	LAS and LSAS as Splitting Schemes	160
4.5.3	Locally Semi-Implicit Scheme for Isotropic Nonlinear Diffusion	161
4.5.4	Proof of the Error Bound for the Exp-Int Approximation	161

<i>Contents</i>	5
5 Spatially Discrete Analysis of Structure-Enhancing Processes	169
5.1 Shock Filters in 1D	170
5.1.1 Semidiscrete Case	172
5.1.2 Fully Discrete Case	179
5.1.3 Equivalence to Local Mode Filtering	184
5.2 Stabilised Inverse Linear Diffusion	186
5.2.1 Semidiscrete Analysis	188
5.2.2 Numerical Tests	199
5.3 Forward-and-Backward Diffusion	204
5.3.1 Stability and Well-Posedness Analysis	205
IV Variational Image Deconvolution	211
6 Variational Models and Integro-Differential Equations for Image Deconvolution	213
6.1 Introduction to Basic Deblurring Concepts	213
6.1.1 Basic Blur Model	214
6.1.2 Classification of Deblurring Approaches	215
6.1.3 Related Work	217
6.1.4 Our Contributions	220
6.2 Energy Functionals and Diffusion-Reaction Equations	222
6.2.1 Energy Functionals and Diffusion-Reaction Equations	222
6.2.2 Variational Gradient Descent	223
6.3 Energy Functionals and Corresponding Diffusion-Reaction Equations	224
6.3.1 Data Terms – Reaction Terms	224
6.3.2 Regularisers – Diffusion Terms	227
6.4 Special Model Aspects	231

6.4.1	Outliers and Violation of Model Assumptions	231
6.4.2	Boundary Treatment	231
6.4.3	Choice of Regularisation Weight	236
6.5	Numerical Aspects	238
6.5.1	Discretisation of the Reaction Term	239
6.5.2	Discretisation of the Diffusion Term	240
6.5.3	Stability Considerations	243
6.6	Experiments	245
6.6.1	Deblurring of Synthetically Blurred Images	245
6.6.2	Deblurring of Photographs Blurred During Exposure	248
7	Inequality Constraints in Variational Deconvolution	253
7.1	Basic Framework	254
7.1.1	Constraints for Greyvalue Images	254
7.1.2	Separable Constraints for Multi-Channel Images	258
7.2	Geometric Reinterpretation	261
7.3	Positive Definiteness as an Inequality Constraint	262
	Concluding Remarks	265
	Bibliography	267

Introduction

This monograph is concerned with mathematical models and algorithms for the processing of digital images. It is a cross-section through the author's work in this field over a period of approximately five years. The author's activities during this time have been directed to a number of different problems, such as nonlinear methods for matrix-valued image denoising, space-discrete analysis of PDE-based image filters and design of new numerical algorithms for such PDE filters, differential geometric models, image sharpening, generalised morphological scale spaces etc.

Although not all of these appear within this work, the three dominant topic areas have been included: median filters and M-smoothers for multi-channel data, space-discrete analysis and numerical schemes for PDE-based image filters, and variational approaches to the sharpening of blurred images. This is already a fairly heterogeneous collection as far as one considers the types of problems, of models, and of results: While multi-channel M-smoothers are modeled and implemented on a fully discrete level, the work on deblurring models presented here consequently follows a space-continuous paradigm. The investigation of multi-channel M-smoothers is focussed on a theoretical foundation and involves axiomatic considerations leading to a definition, and a detailed study of the so obtained concept. In contrast, the emphasis of the work on space-discrete analysis of PDE-based filters is very much on the numerical side of filters that are already well established. In deblurring, instead, the main goal is on the development of models and their comparison on realistic image material.

What unites these topics, nevertheless, are guiding principles that relate to the author's approach to image processing problems in general. As these concepts penetrate the entire work presented, they have also been chosen to entitle it.

Dynamics. Many image filters are formulated as processes that in a sequence of discrete steps or possibly by advancing on a continuous scale successively transform an image from its given initial state towards the desired filtered image. It is natural and common to interpret such a filter as an evolution in (discrete or continuous) time, where the notion of time is that of an artificially introduced parameter, in contrast

to the natural concept of time in image sequences (videos/movies) that is in fact an additional dimension of the image domain.

As time-dependent processes are mathematically modelled by dynamical systems, it is possible to consider also all kinds of image filters that admit such a time-dependent formulation from a dynamical systems point of view. At times this may be a mere reformulation, while at other times it brings about substantial benefits, when qualitative or quantitative results on dynamical systems can be transferred.

Most image filters regarded in this work come in a time-dependent formulation, which makes the dynamical systems viewpoint a continuous theme of the work. It is moved to the foreground specifically in the part on space-discrete analysis and numerics of PDE filters.

The potential of dynamical systems in image processing is not exhausted in the present work. An important direction that is not covered is formed by oscillatory diffusion-reaction systems, see [168, 186].

Geometry. Most images are acquired by the measurement of, or otherwise represent spatially distributed physical quantities. Meaningful processing of images must therefore be consistent with their physical semantics. This firstly implies that images cannot be thought without the geometric structure of a suitable model of space. Secondly, in most cases the variety of data values themselves possesses naturally a geometric structure, which may or may not be interwoven with the geometric structure of the underlying domain. Both aspects imply that a geometric interpretation of image data is valuable in guiding the selection and use of mathematical models. Analytic concepts that allow a sensible processing of image data should also be accessible by geometric interpretation.

A central role in modern geometry is played by invariances. This is also a fruitful approach in image processing.

Mathematics has developed well-developed geometrical frameworks that image processing can capitalise on if it is done on a consequent geometric foundation. Regarding space-continuous models, differential geometry is a particularly adequate choice, see e.g. [175]. A collection of discrete geometric approaches to image processing can be found in [25].

Related work. An impressive literature has emerged on mathematical image analysis mainly during the last two decades. The topics treated in this work refer to different sectors of this big edifice. For this reason, it appears reasonable to place the discussion of related work in the introductory sections of the single chapters.

Thanks and Acknowledgements

It is a pleasant obligation for me to thank all those people whose substantial support was essential for this work.

My first and deepest thanks go to my wife, Veronika Winkler, for sharing all highs and lows with me, for making me feel at home while at home, for taking care of so many things – for being there and for being with me. Next, I thank our son Simeon for filling us with delight throughout the last two years.

Regarding the academic part, I feel highly indebted to Joachim Weickert for giving me the opportunity to enter the scientific area of mathematical image analysis, for his guidance on this way, for contributing by inspiration as well as cooperation to much of the work presented here.

Most of my recent research work has been done in cooperation with colleagues and students in Saarbrücken and other places. For the pleasure of joint work I cordially thank Florian Becker, Tobias Becker, Achim Bergmeister, Susanne Biehl, Michael Breuß, Thomas Brox, Andrés Bruhn, Oliver Demetz, Christian Feddern, Irena Galić, Guy Gilboa, Pilwon Kim, Pavel Mrázek, James G. Nagy, Peter J. Olver, Christoph Schnörr, Gabriele Steidl, David Theis, Joachim Weickert. With many of them, I co-authored research papers during this period. Several of their contributions will also receive special mention later on throughout this work.

I'd like to express my gratitude to Michael Breuß for his encouragement, and for sharing with me his opinion on part of this work.

The exciting working atmosphere in the Mathematical Image Analysis Group in Saarbrücken substantially promoted my work.

I am strongly thankful to James G. Nagy and Emory University for their hospitality during three months in 2005, as well as Peter J. Olver, the Institute for Mathematics and its Applications at the University of Minnesota, Minneapolis, for their hospitality during three months in 2005–06.

It is gratefully acknowledged that part of the research presented here has been supported by Deutsche Forschungsgemeinschaft under grant We 3563/2-1.

Magnetic resonance and diffusion tensor magnetic resonance data sets have been provided by Oliver Gruber and Ilona Henseler, Saarland University Hospital, Homburg, Germany. A tensor-valued data set from a fluid dynamics simulation is owed to Wolfgang Kollmann (UC Davis) and Geric Scheuermann (University of Leipzig). For the visualisation of 3D tensor data the tool `t3v` by Stephan Didas and Luis Pizarro was used.

Structure

This work consists of four parts. While the first part presents some preliminaries which are used continually in the remainder of the book, each of the other three parts focusses on one specific area of my research in image processing.

The introductory **Part I** is divided into two chapters. Chapter 1 collects some basic concepts about digital images and their mathematical representation, focussing on those types of images which will play a role throughout this work. Chapter 2 contains a short overview of diffusion filter models for images, which are used continually during later chapters, and introduces the relevant nomenclature and notations.

Part II with its single Chapter 3 is devoted to M-smoothers for multi-channel data, with emphasis on matrix-valued images. M-smoothers are an interesting, simple non-linear filter class for scalar-valued images that contains the median as one particularly interesting and well-studied example. The extension of the median concept to multi-channel data stands in the centre of the chapter, followed by an investigation of relevant properties and extensions. The generalisation is consequently built on geometric principles: a geometric interpretation of the multi-channel image data, and invariances motivated from the underlying physical models.

More specifically, the multi-channel median definition is based on the minimisation of a geometrically motivated objective function which measures the sum of distances of a variable matrix to the given data matrices. Unlike some other approaches, this approach does not require by definition that the median has to be one of the given data values. Notwithstanding, in many cases this happens to be the case. As a consequence, multi-channel median filters possess root signals similar to their scalar-valued counterparts.

Like their scalar-valued counterparts, multi-channel median filters show excellent capabilities for structure-preserving denoising. This is demonstrated for the matrix-valued situation by experiments on diffusion tensor imaging, fluid dynamics and orientation estimation data. The latter examples give rise to a new variant of a robust adaptive structure tensor.

By generalising the idea of the multi-channel median filters, a variety of other local filters for multi-channel images is designed. These include a one-parameter family of M-smoothers, mid-range filters, but also weighted medians and α -quantiles. Links to fundamental operations of matrix morphology are discussed.

With exception of a short excursion, all these filters are considered in a space-discrete setup. In this case, iterated median filters are a nonlinear iterated function system, thus a time-discrete dynamical system. Some important properties of single- and multi-channel median filters can be translated into qualitative statements about the

corresponding dynamical system: Invariance under geometric transformations translates into symmetries of the dynamical system, while root signals are (nontrivial) steady states of it. In this sense, particularly Chapter 3 integrates dynamic and geometric ideas.

Part III with Chapters 4 and 5 is concerned with time-continuous dynamical systems consisting of ODEs, which form spatial discretisations of PDE-based image filters. When also a further time discretisation is considered, one obtains time-discrete dynamical systems.

This dynamic approach can be exploited in different directions. On one hand, it enables to analyse important theoretical properties of image filters. The transition from a space-continuous PDE to a space-discrete ODE or space-time-discrete iterated function system alleviates the analysis, and at the same time it creates a setup which immediately reflects the way filters are practically used, that is on discrete images. The interactions between pixels are explicitly modelled. On the other hand, it is possible to devise suitable discretisations with desired properties. In particular it can be ensured that features of the continuous models are also present in their discretised counterparts, e.g., conservation properties or invariances.

Chapter 4 is devoted to a space-discrete analysis of linear, isotropic nonlinear and anisotropic nonlinear diffusion filters and the design of a class of numerical schemes with favourable properties for image processing applications. The core idea here is to found the analysis on minimalistic signals and images consisting of two (in the 1D case) or four (in the 2D case) pixels. It is demonstrated that this idea can be applied in great generality to a broad variety of diffusion filters, covering virtually all kinds of diffusion filters that are established in contemporary image processing. Also the popular total variation (TV) flow and similar singular diffusion processes can be treated by this approach, handling their singularities in a better way than in common algorithmic realisations. Moreover, the framework transfers in a clear way to higher-dimensional and multi-channel settings.

Though nonlinear diffusion filters can locally act structure-enhancing, diffusion filters as studied in Chapter 4 are primarily smoothing filters. Chapter 5 instead focusses at the well-posedness analysis of algorithms whose primary purpose is the enhancement of structures and which in particular in their continuous forms are known to suffer from instabilities and ill-posedness. This characterisation adumbrates a more heterogeneous class of methods which can hardly be exhausted by analysis. Therefore three filters are investigated, each of which is exemplary for one type of structure-enhancing PDE filter proposed in image processing literature, either in pure form or as building block in more complex algorithms: firstly, a simple shock filter as proposed by Osher and Rudin; secondly, a stabilised inverse linear diffusion process which can be seen as prototype of inverse diffusion processes used in some advanced numerical methods; and thirdly, the so-called forward-and-backward (FAB) diffusion,

a diffusion process featuring (isotropic) direct and inverse diffusion at different locations. For all three filters, analysis is currently restricted to the 1D situation, reflecting the fact that this research is still at an earlier stage.

Part IV with Chapters 6 and 7 investigates models for the deblurring of images. Blurring is a ubiquitous problem in image acquisition, arising from such different sources as atmospheric perturbations, limitations of the optical (or other sensing) systems, object and camera motion during the image acquisition process. Therefore the removal of blur from acquired images, thus deblurring, is a desideratum in many application fields, creating a strong demand for suitable algorithms.

In simple cases blurring can be modelled as convolution of a sharp image with some convolution kernel. Accordingly, deblurring is also called deconvolution, a term that is also used beyond the mentioned special cases. A great variety of deconvolution methods are proposed in the extensive literature of the field. They differ in their assumptions, means, and generality, but due to the severe ill-posedness of deconvolution problems, no existing method can adequately serve all needs.

Deconvolution methods discussed in this work belong to the class of variational approaches. Their evaluation gives rise to deblurring PDEs. At one point we extend the view slightly to include closely related PDEs which are not strictly derived from variational models. The focus of study is on the variational modelling for deconvolution.

Chapter 6 presents a fairly general framework for variational deconvolution of images with known point-spread function. Its core component are variational functionals that can be chosen convex or non-convex, and which lead to diffusion-reaction PDEs that can act edge-preserving or even edge-enhancing during the sharpening process. Specific ingredients are robust data terms that allow for much more insensitivity with respect to noise, model violations and imprecise estimation of the point-spread function, and non-convex regularisation terms leading to edge-enhancing diffusion of Perona-Malik type. The study also includes tensor-driven anisotropic diffusion terms for regularisation. The boundary treatment and strategies for steering the regularisation weight are discussed. Experiments are presented that allow a visual, and to some degree, quantitative comparison of deconvolution methods.

Chapter 7 is concerned with a useful modification of this framework. It shows how inequality constraints, such as bounds on pixel values, can be incorporated into the variational model. Constraints are modelled either by a reparametrisation of the image values, or in a differential geometric way by modifying the metric on the image values in which a gradient descent is carried out. Particularly the latter formulation allows to realise fairly general constraints on multi-channel images. Experiments on positivity-constrained colour image deconvolution and positive definite deconvolution

lution of matrix-valued image data demonstrate the broad applicability and performance of the model.

While the design of models throughout Part IV is driven by a geometric sight, Chapter 7 adds a different geometric aspect when metrics on the range of image values (and the induced ones on image spaces) are used to enforce inequality constraints. This is a striking example how genuinely geometric ideas can be made fruitful in designing models of immediate practical relevance.

Revision Note

The original version of this work has been submitted to the Department of Mathematics of Saarland University, Saarbrücken, Germany, in November 2007.

Since then, I have made available revisions in July 2013 and February 2016 (this one). In these revisions, I have corrected a few wrongful statements and proofs (as listed below), and several minor mistakes and typos. Many of them have been brought to my attention by the reviewers of the thesis and other readers, which I gratefully acknowledge. I'd like to thank particularly to Mila Nikolova, Joachim Weickert, and Tobias Becker for helpful comments.

The sections where wrongful statements have been corrected are

- Section 4.5: here the original version contained a mistake which has been corrected with substantial new arguments and estimates (revised July 2013);
- Section 3.2.2.3: Statement (b) of Proposition 3.9 was corrected; its proof has been replaced with a reference to a publication that contains a corrected proof (revised February 2016);
- Section 3.5.2: the original version of Proposition 3.15, its supporting statements and the subsequent Corollary 3.16 were flawed by omitting contributions from mixed-derivative terms in the Taylor expansion of the evolving function (provisorially revised February 2016 by removing or restricting wrongful statements; full revision upcoming).

Apart from these sections, the revised version reflects the state of research at the time of submission of the original thesis. Neither new scientific material has been introduced, nor have references been updated with newer literature.

Part I

Preliminaries

Chapter 1

Single- and Multi-Channel Images

In this work, we are concerned with mathematical models and algorithms for the processing of *images*. By an image we mean an arrangement of values of uniform type on a suitable domain. This domain can either be continuous, in which case it is typically some domain in \mathbb{R}^2 or \mathbb{R}^3 , or discrete. In the latter case the most common setting is a regular grid in \mathbb{R}^2 or \mathbb{R}^3 . In practice digital images are necessarily given on a discrete domain, but in most cases this is considered as a subset of a continuous (physical) domain. The measurement process that extracts the data on the discrete domain is called *sampling*.

Though in practice images are available only in sampled form, and digital image processing algorithms need to be devised finally for discrete-domain images, we consciously include the case of a continuous domain. The reason for this is that a discrete image domain is typically obtained by discretising a continuous physical domain, and a variety of mathematical models that are of great use in image processing are formulated on continuous domains. This includes particularly models based on partial differential equations (PDEs). Continuous-domain modelling often facilitates the representation of physically required properties of the data. The most important property of this kind is rotational invariance. Its fulfilment is inherent to any PDE-based model, while fully discrete methods that are popular in nowadays computer vision, like graph-cut methods, often perform poor in this respect.

The range of the image data, on the other hand, can also be continuous, like \mathbb{R}^d or a compact subset of it, or discrete. The process that reduces a continuous image range to a discrete subset such that data values need to be approximated by the available discrete set is called *quantisation*, the approximation error incurred herein is the *quantisation error*. Stored digital images are frequently quantised on integer scales, which leads to a quantisation error that needs consideration. Image processing algorithms, however, can be implemented using float values such that further quantisation errors during the computation are negligible.

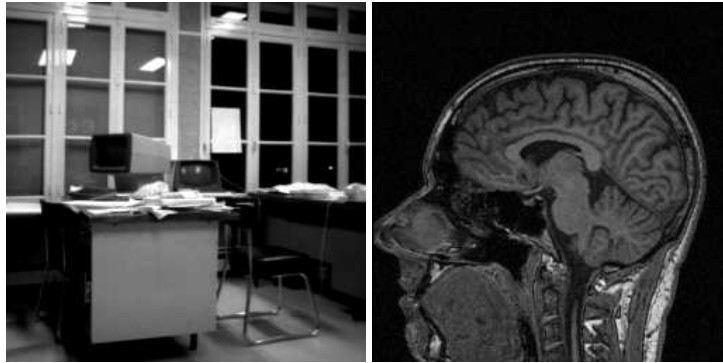


Figure 1.1: Grey-value images. *Left: (a)* Black-and-white photograph (*office image*). *Right: (b)* One slice of a magnetic resonance data set of a human head. MR measurement by O. Gruber, I. Henseler, Saarland University Hospital, Homburg.

We mention that our fairly general definition includes also image types with discrete ranges that are not subsets of continuous ranges (with a meaningful embedding). One example are label maps that assign a label from a finite set to each pixel, without meaningful arithmetic operations in the label set. Many segmentation algorithms (see e.g. [41, 141]) generate this type of images. In the present work, however, these image types will not be considered any further.

1.1 Grey-Value Images

In the simplest type of images the image values are simply brightness values, commonly denoted as grey-values. The prototype of such an image is a digital black-and-white photograph, see Figure 1.1(a).

There exist, of course, digital images with other scalar-valued types of image data. One important example are X-ray densities in digitised X-ray radiography or computerised tomography, which latter application gives rise to three-dimensional image domains. Further we mention magnetic resonance (MR) images in whose most wide-spread basic variant the scalar data encode the density of hydrogen atoms in tissue, modulated by their chemical environments. These, too, arise naturally as 3D data sets. A third example is given by ultrasound images in which the scalar-valued data represent a second directional derivative (in beam direction) of the acoustic impedance.

As arbitrary scalar-valued quantities can be visualised as brightness data, it is common to denote all kinds of scalar-valued images as grey-value images, see Figure 1.1(b).



Figure 1.2: *From left to right*: Colour photograph, 240×320 pixels, and its red, green, and blue channels.

1.2 Colour Images

The visual system of most humans distinguishes a three-dimensional gamut of colours, which can be represented as a region in a three-dimensional space spanned by axes corresponding roughly to red, green, and blue colour perception. Accordingly, also technical systems which acquire and process colour image data, are designed to use three-dimensional colour spaces, with the RGB colour space based on the basic colours red, green and blue as the most prominent representative, see Figure 1.2.

An RGB colour image is mathematically represented by an \mathbb{R}^3 -valued function on its image domain. Often such images are already called *vector-valued*, for a twofold reason. Firstly, in computer science the term vector is commonly used to denote a data structure that represents a tuple of data of equal type. Secondly, the RGB colour space can indeed be equipped with a vector space structure. A caveat about this terminology is that it can be confused with vector fields (see below). We refrain therefore from using this term in the sequel. Instead, our general term for \mathbb{R}^d -valued images independent of additional structures specified on the image range will be *multi-channel images*.

Colour image processing examples in this work (especially Chapters 6 and 7) use also the RGB colour space, notwithstanding the possibility to adapt the methods being discussed to other colour spaces.

1.3 Vector Fields

Though vector fields will not play a role in this work, we shortly mention them as a type of multi-channel images with additional structure. In analogy to the mathematical concept of a vector field we denote a multi-channel image as a vector field if the dimensionality of the image range equals that of the image domain, and the semantics

of the image values is that of vectors in the tangential space of the image domain. For digital images, of course, this means the tangential space of the underlying continuous space.

As a consequence, geometric transformations of the image domain entail changes of image values. For example, if such an image is rotated, one has to rotate its vectorial values in the same sense: The image transforms as a contragredient tensor of first order. This is in clear contrast to, e.g., colour images whose rotation does not affect the image values. One example of this image type are optic flow fields, see e.g. [40, 45, 141].

1.4 Matrix-Valued Images

In contemporary image processing, there is an increasing interest in processing *second-order tensor fields*. They appear as physical quantities which are measured e.g. by *diffusion tensor magnetic resonance imaging (DTMRI)* [163], or computed, as in computational fluid dynamics, or as derived quantities like the *structure tensor* [91] which plays an important role in fields like motion detection, texture analysis or segmentation [27, 97, 139, 169]. Fields of application also include geophysics, material science and civil engineering.

Sampled second-order tensor fields which are obtained either from measurement or computation come in the form of matrix-valued images, i.e., images whose values are symmetric matrices. Similar to the previous discussion about vector fields it should be noted that “matrix-valued” and “second-order tensor” aren’t synonymous – a tensor field requires that the matrix indices refer to a coordinate basis of the image domain space. Thus, tensors have a characteristic transformation behaviour: If the image is rotated, the matrix values must be transformed accordingly.

While it is possible to construct matrix-valued images without this tensor semantics, matrix-valued images in this work will in fact always be representations of tensor fields. In particular, diffusion tensor magnetic resonance imaging (DTMRI) will occupy a broad space; therefore the following three subsection will give a brief description of DTMRI data and two methods to visualise these which will be used in this work.

This work can neither give a full account of DTMRI measurement techniques, nor of the physical background, nor of all current visualisation techniques. More information can be found in the edited book [213] which gives a broad overview over tensor field processing and visualisation techniques, and the references therein.

1.4.1 Diffusion Tensor Images

Magnetic resonance imaging (MRI) in its most widespread form is basically a technique to detect water molecules (more precise, hydrogen nuclei, i.e., protons) with spatial resolution. By a modification of the measurement technique – *diffusion weighted magnetic resonance imaging (DWMRI)* – it allows even to determine the rate at which water molecules at a certain location diffuse, projected to some direction.

Repeating this measurement for different direction (under the assumption that the diffusion properties do not change between measurements) enables the construction of an angularly resolved representation of the diffusion behaviour.

The simplest model for the anisotropic diffusion of water molecules assumes a random Brownian motion in which the motion vectors of a molecule during subsequent time intervals are independent and identically distributed, and this property remains valid for any refinement of the subdivision of the time axis into sub-intervals. Under this assumption, it is clear that the probability density of motion vectors for a time interval of duration τ is necessarily a three-dimensional Gaussian. Equivalently, this probability density function describes the spatial distribution of water molecules initially located at the origin $(x, y, z) = (0, 0, 0)$ after time τ . This Gaussian distribution is given by the equation

$$p(\mathbf{x}) = \frac{1}{8(\pi\tau)^{3/2}(\det D)^{1/2}} e^{-\frac{\mathbf{x}^T D^{-1} \mathbf{x}}{4\tau}} \quad (1.1)$$

where $\mathbf{x} \in \mathbb{R}^3$, and D is the covariance matrix of the Gaussian distribution, cf. [163, 2]. The latter is a positive definite symmetric 3×3 matrix which is characterised by its transformation behaviour under Euclidean transformations as a second-order tensor. It is denoted in this context as *diffusion tensor* [163].

In a DWMRI measurement, a special MR sequence is used to record the average displacement of water molecules within each voxel in a particular direction due to diffusion over a short time interval (approx. 10 ms). Up to a scalar factor depending on the diffusion time, the result is a value

$$b(\mathbf{e}) = b_0 w(\mathbf{e}) = b_0 \cdot \mathbf{e}^T D \mathbf{e} \quad (1.2)$$

where \mathbf{e} is a unit vector denoting the diffusion direction measured, b_0 is the proton density of the voxel measured, and D the diffusion matrix. The value $w(\mathbf{e})$ is called *directional diffusivity*.

To measure diffusion tensors by magnetic resonance, six directional diffusion-weighted measurements are required, together with the direction-insensitive proton density measurement b_0 for reference. From these data, the six independent entries of D can be computed. A standard choice for the six diffusion directions to be measured

is given by the unit vectors

$$\begin{aligned} e_1 &= \frac{1}{\sqrt{2}}(1, 1, 0)^T, & e_2 &= \frac{1}{\sqrt{2}}(1, 0, 1)^T, & e_3 &= \frac{1}{\sqrt{2}}(0, 1, 1)^T, \\ e_4 &= \frac{1}{\sqrt{2}}(1, -1, 0)^T, & e_5 &= \frac{1}{\sqrt{2}}(1, 0, -1)^T, & e_6 &= \frac{1}{\sqrt{2}}(0, 1, -1)^T \end{aligned} \quad (1.3)$$

which are symmetrically arranged on the sphere S^2 , thereby providing an equally spaced angular sampling. Given the diffusivities $w_i := w(e_i), i = 1, \dots, 6$ measured in these six directions, the matrix components of $D = (d_{ij})_{i,j=1,2,3}$ can be computed via

$$\begin{aligned} d_{11} &= \frac{1}{2}(w_1 + w_2 - w_3 + w_4 + w_5 - w_6) \\ d_{22} &= \frac{1}{2}(w_1 - w_2 + w_3 + w_4 - w_5 + w_6) \\ d_{33} &= \frac{1}{2}(-w_1 + w_2 + w_3 - w_4 + w_5 + w_6) \\ d_{12} &= d_{21} = \frac{1}{2}(w_1 - w_4) \\ d_{13} &= d_{31} = \frac{1}{2}(w_2 - w_5) \\ d_{23} &= d_{32} = \frac{1}{2}(w_3 - w_6). \end{aligned} \quad (1.4)$$

It is worth mentioning that neither the six matrix entries, nor even the diffusivity components in three orthogonal directions are measured independently during this process. In the matrix representation, measurement noise will therefore display inter-channel correlation. For a more detailed discussion of noise in DTMRI images, see [101].

1.4.2 Tiled Grey-Value Visualisation

It is beyond the scope of this work to discuss visualisation aspects in an exhaustive manner. Much intensive research has been done in this field, see e.g. [200, 22] and further references therein. This and the following subsection are dedicated to describe in compact form the visualisations that will be used in this work.

One way to visualise matrix-valued images in 2D is to show each channel, i.e., each matrix entry, in a separate grey-value image, and to arrange these grey-value images into one frame in the same way as the corresponding matrix entries are arranged in a matrix. Thereby, each sub-image has the pixel dimensions of the entire matrix-valued image.

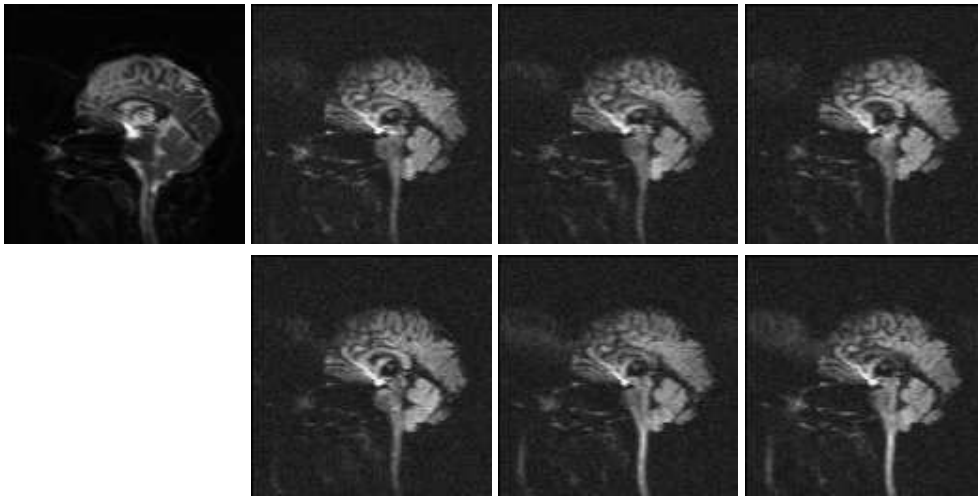


Figure 1.3: Diffusion weighted data measured in DTMRI imaging. *Top left:* b_0 data for one planar (sagittal) section (127×127 voxels). *Other frames:* Six directional diffusion images b_i for the same section and the diffusion directions from (1.3). The amplitude of data in b_0 is about 6.37 times as wide as that in b_1, \dots, b_6 . For better visibility, the greyvalue conversion scale for the latter has therefore been stretched by the same factor. Measurements by O. Gruber, I. Henseler, Saarland University Hospital, Homburg.

Since positive and negative matrix entries need to be represented, a middle grey-tone is chosen to represent zero; in our visualisation, this value is fixed to 128 out of a range $[0, 255]$.

Note that in this representation the upper right and lower left sub-images are equal because of the symmetry of the matrices. For positive semidefinite tensor data such as DTI, the main diagonal entries in the upper left and lower right sub-image contain only nonnegative values while the off-diagonal entries can be of either sign but have smaller variation. While this visualisation is simple and can be generated straightforward from the data, it is negligent of the physical and geometric properties of matrix-valued data that represent tensors. Anisotropy of the tensors is not represented in a uniform and easy-to-interpret way in these images, and rotating a tensor in space changes the corresponding pixel in the tile images in a complex and non-intuitive manner.

1.4.3 Ellipsoid Visualisation

An alternative way to visualise symmetric positive semidefinite matrix images that is more adequate to the physical and geometric semantics of second order tensors is to use ellipses (in 2D) or ellipsoids (in 3D). Actually, this is a special case of *glyph-based* visualisation, see [22]. Each glyph (ellipse or ellipsoid) represents the matrix value of

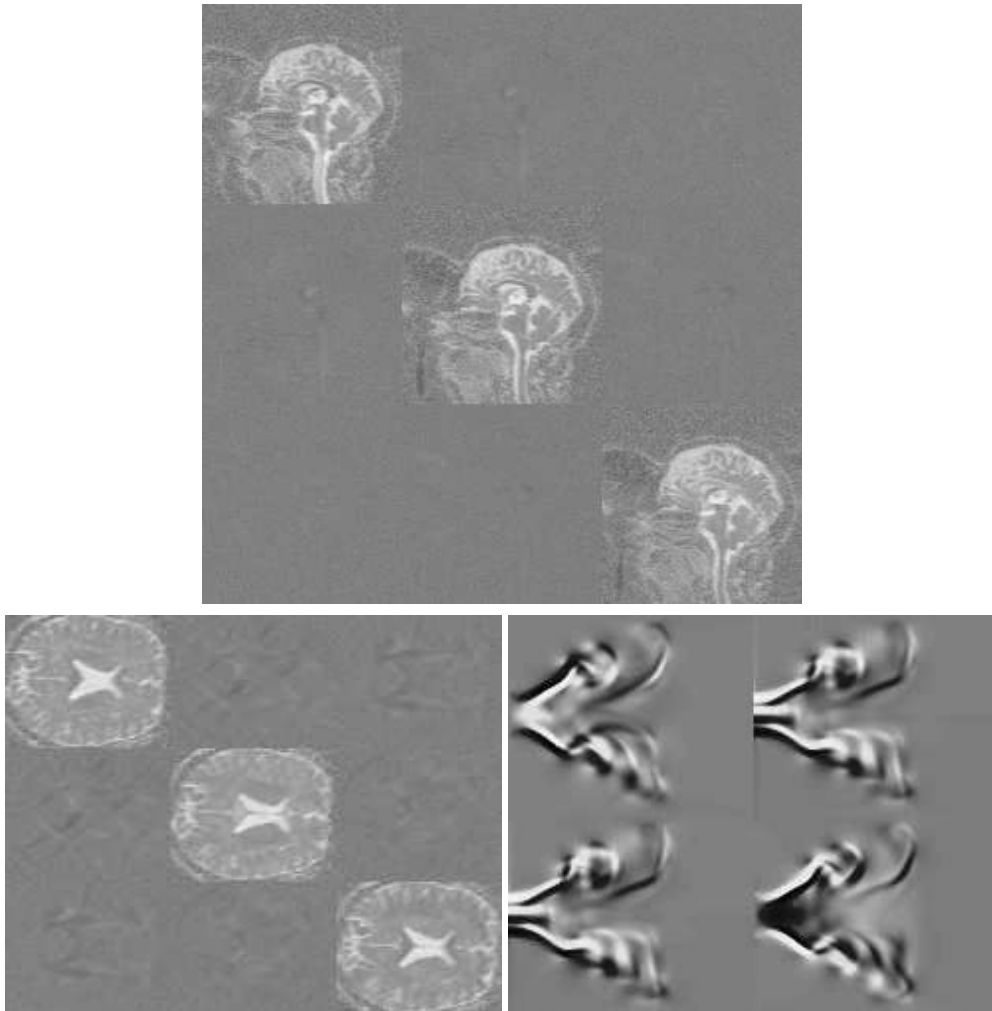


Figure 1.4: Tiled grey-value visualisation of matrix-valued images. *Top: (a)* Sagittal slice (127×127 voxels) of DTMRI data computed from the diffusion-weighted images shown in Figure 1.3. Structures outside the skull contours are aliasing artifacts caused by the specific scanning parameters. *Bottom left: (b)* One 82×66 voxel horizontal (transversal) slice excised from the same DTMRI data set, slightly above eye level (looking to the right). The selected plane cuts through the ventricles (visible as bright structure in the middle of the main diagonal tiles) and the corpus callosum (in the hollow of the fork-like endings of the ventricle section to the left and right). The folded cortex structure is also visible. *Bottom right: (c)* 2D data set (124×101) of 2×2 matrices representing computed deformation tensors from a fluid dynamics simulation. These matrices are symmetric but can have positive and negative eigenvalues.

a single pixel or voxel. The exact correspondence is discussed in more detail below, together with some useful modifications.

In the 2D case, ellipses are rendered directly; in contrast, ellipsoids representing 3×3 matrices need to be shown in some shaded perspective view to make their 3D orientation and shape recognisable.

A visualisation of this type allows a direct visual interpretation: Anisotropy can immediately be judged from the shape of the glyph, while rotation of a matrix is represented by the same rotation applied to the glyph.

A disadvantage is that the representation of glyphs in a size that allows to recognise their geometric features clear enough imposes tight limits on resolution. Only image regions with diameters of some tens of pixels can be displayed in a reasonable way. In 3D data sets, even tighter limitations apply in depth direction since already a few layers of ellipsoids arranged one behind the other lead to visual clutter in the perspective representation. Animations can alleviate this problem but are of course not feasible in print. In this work, we will therefore mostly restrict display to single layers.

1.4.3.1 Standard Representation

In all cases, a positive symmetric matrix is represented by an ellipse or ellipsoid whose principal axis *directions* coincide with the eigenvector directions of the matrix.

In the simplest case, the *lengths* of the principal axes are just the eigenvalues (in a suitable scaling).

Mathematically, this means that the positive symmetric matrix A is represented by the ellipse or ellipsoid with the equation

$$x^T A^{-2} x = 1. \quad (1.5)$$

By enabling a direct visual comparison of eigenvalues, this representation supports the judgement of anisotropy of single diffusion tensors.

1.4.3.2 Isosurface Representation

Looking specifically at the case of diffusion tensors $A = D$ whose physical meaning is characterised by the anisotropic diffusion equation (1.1), it becomes clear that the visual representation can be brought into a closer relation to the physics of diffusion if the equation

$$x^T A^{-1} x = 1 \quad (1.6)$$

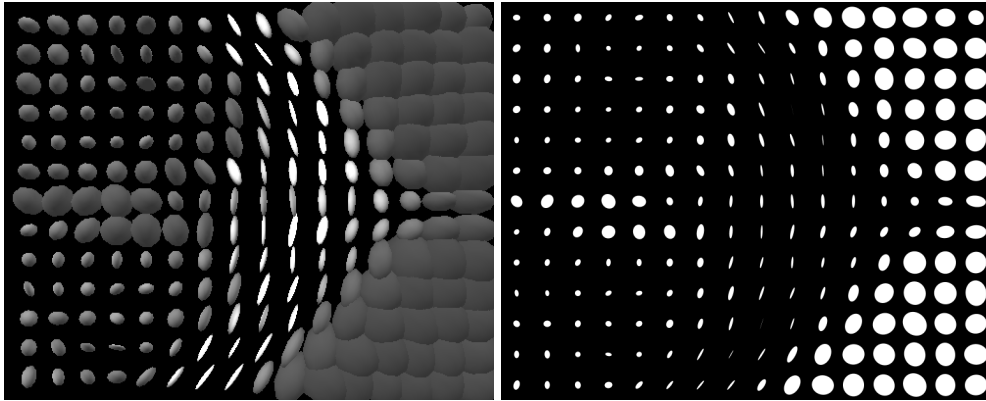


Figure 1.5: Glyph-based visualisation of positive definite matrix-valued images. *Left: (a)* Detail from the corpus callosum region in Figure 1.4(b), with 3×3 matrices visualised by ellipsoids. *Right: (b)* A 2×2 -tensor data set has been generated from the top left four entries of the DT-MRI matrix data, which refer to the tensor components within the section plane. The 2×2 matrices from the same region as before are visualised by ellipses.

is used to generate glyphs instead of (1.5). In this case, for example, the ellipsoid representing a 3×3 diffusion tensor depicts an isosurface for the probability distribution of the location of water molecules starting at 0 after a short diffusion time.

The principal axis lengths of such a glyph correspond to the *square roots* of the eigenvalues of the matrix represented.

While the immediate physical interpretability of the glyphs is advantageous, the anisotropy of diffusion tensors appears less emphasised in this representation, see Figure 1.6(a). For visualisation purposes, preference is therefore given to the previously mentioned model.

1.4.3.3 Range-Compressed Representations

Another disadvantage of the standard elliptic glyph visualisation is that sets of matrices with large variations in the magnitude of their eigenvalues are difficult to represent. Depending on the scaling, either glyphs representing small eigenvalues are shrunk too much, or glyphs representing large eigenvalues begin to overlap with their neighbours.

This situation typically occurs in DT-MRI images of brain structures adjacent to ventricles.

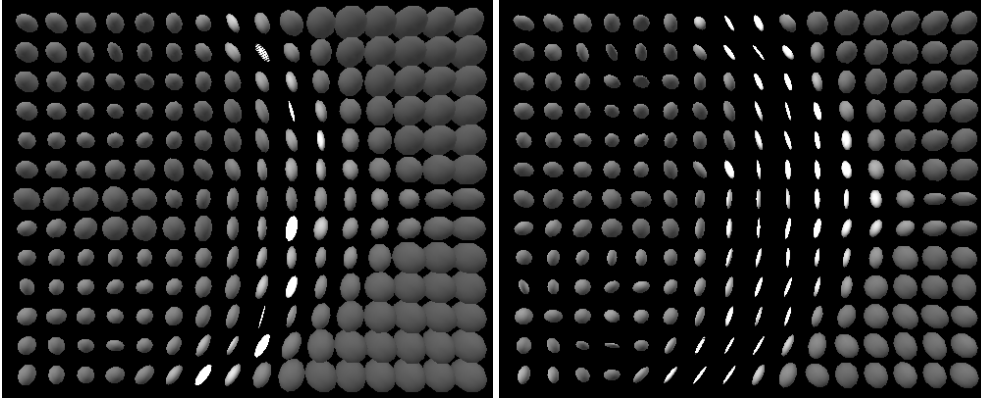


Figure 1.6: Glyph-based visualisation of positive definite matrix-valued images. *Left: (a)* Representation from Figure 1.5(a) repeated. Same set of matrices as in Figure 1.5(a), represented by iso-surfaces. Notice the much less pronounced representation of anisotropy. *Right: (b)* Range-compressed visualisation: Glyphs from (a) are rescaled such that their largest principal axes correspond to 0.333-th powers of largest eigenvalues.

The isosurface representation would alleviate this problem but at the cost of making anisotropy less visible. Alternatively, one could think of masking out or explicitly shrinking the large eigenvalues in ventricle regions.

Another approach is a *range compression*. Here, the glyphs of the standard representation are shrunk or enlarged such that their differences in size are reduced. One possibility is to base this resizing on volume. In this case one would for example scale glyphs such that their volumes correspond no longer to their determinants but only to the γ -th power of it, $0 \leq \gamma < 1$.

The glyph for the symmetric positive definite $d \times d$ -matrix A is then given by the equation

$$(\det A)^{\frac{2}{d}(1-\gamma)} x^T A^{-2} x = 1. \quad (1.7)$$

A difficulty with this approach is that the glyphs of highly anisotropic matrices with one or two eigenvalues close to zero are heavily enlarged and tend to clutter with neighbouring glyphs. Singular matrices can not at all be represented. To overcome these problems, one can also rely on the maximal eigenvalue for resizing. Glyphs are then scaled such that their largest principal axes are proportional to the γ -th powers of the largest eigenvalues, $0 < \gamma < 1$. In this case, the glyph for matrix A is given by

$$(\|A\|_{(\infty)})^{2(1-\gamma)} x^T A^{-2} x = 1. \quad (1.8)$$

We show an example for range compression using (1.8) in Figure 1.6(b).

Since the shrinkage or enlargement acts isotropically, anisotropy remains clearly visible. At the same time, the absolute size of glyphs varies less within the image (even

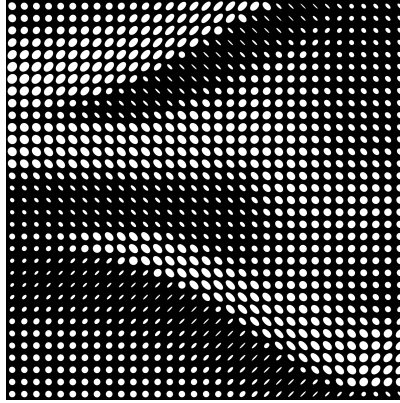


Figure 1.7: A 32×32 region from the 2D tensor field from Figure 1.4(c) represented by ellipses. Since the matrices have positive and negative eigenvalues, they are represented as deformations of a unit disc.

not at all for $\gamma = 0$). What is lost comparing to the standard representation is the possibility for direct visual comparison of eigenvalues of different matrices, but this may be among the least important features in the inspection of the data.

1.4.3.4 Deformation-Based Representation for Indefinite Matrices

Matrix-valued images that contain indefinite or negative definite matrices do not admit an ellipsoid visualisation as discussed so far. However, the idea of this visualisation can be extended to symmetric matrices with eigenvalues of arbitrary sign by using a different motivation from physics: If we interpret these matrices as stress-strain tensors – which can have positive and negative eigenvalues, corresponding to strain and stress, respectively – , we can show their deformation effects on probe objects, like isotropic balls. A spherical probe of unit radius is then deformed by the forces corresponding to the stress-strain tensor A into an ellipsoid whose principal axis directions coincide with those of A , with principal axis lengths $1 + \varepsilon \lambda$ where λ is the corresponding eigenvalue of A and ε a small positive constant that represents deformability of the material. Thus, positive eigenvalues of A express expansion, negative ones compression.

For our visualisation, this means that we simply replace A by $\tilde{A} := I + \varepsilon A$ and use for \tilde{A} the same glyphs as in Fig. 1.5. Note that in this case, it is in accordance with the physical model to use (1.5). Of course, the physical deformation model holds only for small deformations, i.e., small ε . For visualisation, however, one will prefer stronger deformations, just small enough that the eigenvalues of all \tilde{A} in a data set are bounded sufficiently far away from zero.

Chapter 2

Diffusion Filter Models for Single- and Multi-Channel Images

In the subsequent chapters of this work we will be concerned in several ways with diffusion-based image processing methods. The purpose of this chapter is therefore to review the basic concepts and notations around diffusion, introducing them in the form that will be used later.

Diffusion is originally a physical process that equilibrates a physical quantity within some region. This quantity can be, e.g., the concentration of some substance, or heat energy.

Its use in image processing [206] goes back at least to the work of Iijima [110, 111] and serves mainly the goals of denoising and structure simplification. Here, grey values or colour values take the role of the quantities being equilibrated.

The diffusion process is modulated by the rate at which the evolving quantity is transferred at each place and time, which is generally determined by properties of the medium in which diffusion takes place. Since the diffusion process consists in a redistribution of the quantity being regarded, the total amount of this quantity within the considered domain cannot change except due to transport across the domain boundaries. This means, the total amount of the quantity is conserved. In physics, this conserved quantity can be the mass of the substance, or the total heat energy within the volume. In image processing, it will usually be the average grey value.

2.1 Linear Diffusion

Mathematically, diffusion is described by a partial differential equation called *diffusion equation* or, alluding to its second important context, *heat equation*. We start by

considering the simplest case, in which the diffusion is homogeneous – i.e., it does not vary across space – and isotropic – i.e., it shows no variation with spatial direction.

2.1.1 Basic Model

A homogeneous isotropic diffusion process in \mathbb{R}^d is described by a partial differential equation, the *linear diffusion equation*

$$\partial_t u(x, t) = \Delta u(x, t) \quad (2.1)$$

for a function $u : \Omega \times [0, \infty) \rightarrow \mathbb{R}$ on a spatial domain $\Omega \subset \mathbb{R}^d$, with initial conditions

$$u(x, 0) = f(x) \quad (2.2)$$

where f is a given function on Ω , and suitable boundary conditions on $\partial\Omega \times [0, \infty)$. In (2.1), Δ denotes the Laplace operator in the d spatial dimensions.

It is useful for our further considerations to decompose (2.1) into the two first-order differential equations it actually results from: firstly, *Fick's Law*

$$j = \nabla u \quad (2.3)$$

which describes how the gradient of u , i.e., the local differences in the quantity u , gives rise to a *flux* j (a vector field), and secondly, the *continuation equation*

$$\partial_t u = \operatorname{div} j \quad (2.4)$$

which describes the changes in the distribution of values of u that take place if the flux j acts as a transport of this quantity.¹ The conservation of mass or average grey value under linear diffusion is in fact a consequence of (2.4).

2.1.2 Variational Interpretation

Linear diffusion can be considered as a gradient descent for the energy functional

$$E[u] = \frac{1}{2} \int_{\Omega} |\nabla u|^2 \, dx \quad (2.5)$$

whereby a link between diffusion and variational regularisation approaches is established. The gradient descent property can easily be verified by calculating the variation of E . For similar derivations in image processing context see [120, 1.1].

¹The sign convention used here is rather common in mathematical literature. In the physical literature, j is usually taken with opposite sign, leading to minus signs in both equations (2.3) and (2.4).

Variation of E . Let us assume that $w = u + \varepsilon v$ is substituted for u in E , where v is a smooth function on Ω which vanishes on the boundary. We calculate

$$\begin{aligned}
\left. \frac{d}{d\varepsilon} E[w] \right|_{\varepsilon=0} &= \left. \frac{d}{d\varepsilon} \left(\frac{1}{2} \int_{\Omega} |\nabla w|^2 dx \right) \right|_{\varepsilon=0} \\
&= \frac{1}{2} \int_{\Omega} \sum_i \frac{d}{d\varepsilon} (\partial_{x_i} w)^2 \Big|_{\varepsilon=0} dx \\
&= \frac{1}{2} \int_{\Omega} \sum_i \frac{d}{d\varepsilon} (\partial_{x_i} u + \varepsilon \partial_{x_i} v)^2 \Big|_{\varepsilon=0} dx \\
&= \frac{1}{2} \int_{\Omega} \sum_i \frac{d}{d\varepsilon} \left((\partial_{x_i} u)^2 + 2\varepsilon \partial_{x_i} u \cdot \partial_{x_i} v + (\partial_{x_i} v)^2 \right) \Big|_{\varepsilon=0} dx \\
&= \int_{\Omega} \sum_i \partial_{x_i} u \cdot \partial_{x_i} v dx \\
&= \sum_i \int_{\Omega} \partial_{x_i} u \cdot \partial_{x_i} v dx .
\end{aligned} \tag{2.6}$$

Using integration by parts we obtain

$$\int_{\Omega} \partial_{x_i} u \cdot \partial_{x_i} v dx = - \int_{\Omega} \partial_{x_i} (\partial_{x_i} u) \cdot v dx \tag{2.7}$$

and therefore the first variation of E

$$\left. \frac{d}{d\varepsilon} E[u + \varepsilon v] \right|_{\varepsilon=0} = - \int_{\Omega} \Delta u \cdot v dx . \tag{2.8}$$

The *variational gradient* (also called variational derivative or Gateaux derivative) which plays the role of the gradient of E w.r.t. u in the function space from which u is chosen, is defined as the function g for which (2.8) equals $\langle g, v \rangle$ for each admissible function v , with respect to the inner product $\langle \cdot, \cdot \rangle$ of the underlying function space.

With the L^2 inner product, we need only to drop the integral and multiplication by v and find the variational derivative

$$\frac{\delta}{\delta u} E[u] = -\Delta u . \tag{2.9}$$

Gradient descent. With the variational gradient, a gradient descent for the energy functional $E[u]$ is given by

$$\partial_t u = - \frac{\delta}{\delta u} E[u] \tag{2.10}$$

which, by inserting (2.9), becomes exactly (2.1).

2.1.3 Multi-Channel Linear Diffusion

A straightforward generalisation of the energy functional (2.5) to multi-channel images $u = (u_k)_{k \in \Gamma}$ is given by

$$E[u] = \frac{1}{2} \int_{\Omega} \sum_{k \in \Gamma} |\nabla u_k|^2 \, dx \quad (2.11)$$

from which one easily obtains as gradient descent

$$\partial_t u_k(x, t) = \Delta u_k(x, t), \quad (2.12)$$

i.e., channel-wise linear diffusion.

2.2 Isotropic Nonlinear Diffusion

A more general class of diffusion processes is obtained by dropping the assumption of homogeneity, while still retaining the isotropy requirement. In physics, this would mean to consider a medium in which the transport of heat or the diffusing substance takes place at different rates depending on the location, but still at equal speed in all directions at a given place. To reflect this, equation (2.1) is modified into

$$\partial_t u = \operatorname{div}(g(x) \nabla u) \quad (2.13)$$

where g , the *diffusivity*, describes the spatially variant rate of diffusion. As for linear diffusion, the product $j := g(x) \nabla u$ is called *flux*.

By (2.13), an *inhomogeneous linear diffusion* process is described. If the rate of diffusion g is not simply a function of space but is modulated by the diffusing quantity itself, the diffusion process becomes nonlinear.

In image processing, the primary motivation to consider such processes is to improve the preservation of discontinuities, i.e. edges. While in special contexts inhomogeneous linear diffusion processes may be used, the main candidate for this purpose is isotropic nonlinear diffusion with the diffusivity determined by the local gradient, which we will therefore concentrate on in what follows.

2.2.1 Basic Model

We consider the *isotropic nonlinear diffusion* [161, 54, 206] governed by the equation

$$\partial_t u = \operatorname{div}(g(|\nabla u_{\sigma}|^2) \nabla u) \quad \text{on } \Omega \times (0, \infty) \quad (2.14)$$

where the diffusivity function $g(|\nabla u_\sigma|^2)$ is a nonincreasing nonnegative function of the (squared) gradient magnitude, and u_σ denotes a smoothed image $u_\sigma := K_\sigma * u$ with a Gaussian K_σ (or $u_\sigma = u$ for $\sigma = 0$).

Initial and boundary conditions for (2.14) are chosen as for (2.1).

A popular choice for g is

$$g(s^2) = \frac{1}{1 + s^2/\lambda^2} \quad (2.15)$$

introduced in [161] which leads to the so-called *Perona–Malik diffusion*. Other prominent examples include the so-called *Charbonnier diffusivity* [62, 63]

$$g(s^2) = \frac{1}{\sqrt{1 + s^2/\lambda^2}} \quad (2.16)$$

and *Weickert diffusivity* [206]

$$g(s^2) = \begin{cases} 1, & s^2 = 0, \\ 1 - \exp\left(\frac{-3.31488}{(s/\lambda)^8}\right), & s^2 > 0. \end{cases} \quad (2.17)$$

Terminology: Isotropic versus anisotropic. Concerning terminology, we mention that in part of the literature these processes are already called *anisotropic*; see e.g. [161]. This can be motivated by a specific decomposition of the diffusion PDE. To see this, assume that the image domain is two-dimensional, and consider a process without pre-smoothing, i.e., with $\sigma = 0$. Let η denote the unit vector $\nabla u / |\nabla u|$, and ξ a unit vector perpendicular to η . Then (2.14) can be rewritten as

$$\partial_t u = g(|\nabla u|^2) \cdot \partial_{\xi\xi} u + (g(|\nabla u|^2) + 2|\nabla u|^2 g'(|\nabla u|^2)) \cdot \partial_{\eta\eta} u. \quad (2.18)$$

In this representation with respect to a local coordinate frame aligned to the gradient direction, the process is the combination of diffusion in gradient direction (represented by $\partial_{\eta\eta} u$) and perpendicular to it, i.e., in level line direction (represented by $\partial_{\xi\xi} u$). The different intensities g and $g + 2|\nabla u|^2 g'$ of these two components motivate to call them anisotropic.

In contrast, we prefer to follow the terminology established in [206] and many other works that uses the divergence form of a diffusion process to determine whether it is isotropic or anisotropic. We stick therefore with the name *isotropic diffusion* as long as the diffusivity function g is simply a scalar and not dependent on directions; in particular, the resulting flux is aligned with the gradient direction ∇u everywhere. In contrast, the notion of anisotropic diffusion is reserved for processes driven by matrix-valued diffusion tensors, see 2.3.

Forward versus backward diffusion. The representation (2.18) is useful in another way: Though one usually requires g to be nonnegative (compare, however, 5.3 for a process with negative diffusivity), the weight $g(|\nabla u|^2) + 2|\nabla u|^2 g'(|\nabla u|^2)$ of the diffusion in gradient direction can take negative values. Thus, one has always ordinary (forward) diffusion in level line direction while a time-reverted (backward) diffusion might act in gradient direction. A closer look reveals that

$$g(|\nabla u|^2) + 2|\nabla u|^2 g'(|\nabla u|^2) = \frac{d}{ds}(g(s^2) \cdot s) = j'(s), \quad (2.19)$$

i.e., the sign of the derivative of the flux w.r.t. the local gradient determines whether forward or backward diffusion takes place in gradient direction.

For example, the Perona-Malik diffusivity (2.15) leads to $j'(s) = \frac{1-s^2/\lambda^2}{(1+s^2/\lambda^2)^2}$. Thus, pure forward diffusion is encountered in regions with small gradients $|\nabla u| < \lambda$, while backward diffusion occurs where $|\nabla u| > \lambda$.

2.2.2 Variational Interpretation

Like linear diffusion, also the isotropic nonlinear diffusion process (2.14) can be represented as gradient descent for an energy under some conditions: Provided that the diffusivity g depends on an unsmoothed gradient magnitude $|\nabla u|$ (i.e., the pre-smoothing parameter σ is set to zero), and $g \equiv \Psi'$ for a monotonically increasing differentiable function $\Psi : \mathbb{R}_0^+ \rightarrow \mathbb{R}$, we can consider the energy functional

$$E[u] = \frac{1}{2} \int_{\Omega} \Psi(|\nabla u|^2) dx. \quad (2.20)$$

The function Ψ can be interpreted as a *penaliser function* that determines how much some gradient magnitude encountered at a point in the image domain contributes to the energy E .

We will show that (2.20) possesses the gradient descent

$$\partial_t u = \operatorname{div}(\Psi'(|\nabla u|^2) \nabla u), \quad (2.21)$$

i.e., (2.14) with $g \equiv \Psi'$ and $\sigma = 0$. To this end, we will calculate again the variation of $E[u]$.

Variational derivative calculation. Substitute $w = u + \varepsilon v$ for u in E , where v is assumed to be a smooth function on Ω which vanishes on the boundary. We have then

$$\left. \frac{d}{d\varepsilon} E[w] \right|_{\varepsilon=0} = \left. \frac{d}{d\varepsilon} \left(\frac{1}{2} \int_{\Omega} \Psi(|\nabla w|^2) dx \right) \right|_{\varepsilon=0}$$

$$\begin{aligned}
&= \frac{1}{2} \int_{\Omega} \Psi'(|\nabla u|^2) \cdot \sum_i \frac{d}{d\varepsilon} (\partial_{x_i} w)^2 \Big|_{\varepsilon=0} dx \\
&= \frac{1}{2} \int_{\Omega} \Psi'(|\nabla u|^2) \cdot \sum_i \frac{d}{d\varepsilon} (\partial_{x_i} u + \varepsilon \partial_{x_i} v)^2 \Big|_{\varepsilon=0} dx \\
&= \frac{1}{2} \int_{\Omega} \Psi'(|\nabla u|^2) \cdot \sum_i \frac{d}{d\varepsilon} \left((\partial_{x_i} u)^2 + 2\varepsilon \partial_{x_i} u \cdot \partial_{x_i} v + (\partial_{x_i} v)^2 \right) \Big|_{\varepsilon=0} dx \\
&= \int_{\Omega} \Psi'(|\nabla u|^2) \cdot \sum_i \partial_{x_i} u \cdot \partial_{x_i} v dx \\
&= \sum_i \int_{\Omega} \Psi'(|\nabla u|^2) \cdot \partial_{x_i} u \cdot \partial_{x_i} v dx .
\end{aligned} \tag{2.22}$$

Integration by parts yields

$$\int_{\Omega} \Psi'(|\nabla u|^2) \cdot \partial_{x_i} u \cdot \partial_{x_i} v dx = - \int_{\Omega} \partial_{x_i} \left(\Psi'(|\nabla u|^2) \cdot \partial_{x_i} u \right) \cdot v dx \tag{2.23}$$

and therefore

$$\frac{d}{d\varepsilon} E[u + \varepsilon v] \Big|_{\varepsilon=0} = - \int_{\Omega} \operatorname{div} \left(\Psi'(|\nabla u|^2) \nabla u \right) \cdot v dx \tag{2.24}$$

Dropping as before the integral and multiplication by v , we arrive at the variational gradient

$$\frac{\delta}{\delta u} E[u] = - \operatorname{div} \left(\Psi'(|\nabla u|^2) \nabla u \right). \tag{2.25}$$

Inserting (2.25) into (2.10) leads exactly to (2.21), which proves our assertion.

By (2.20), nonlinear isotropic diffusion is also linked to non-quadratic regularisation methods [62, 63, 154, 181, 182].

2.2.3 Role of Pre-Smoothing

While the derivation by gradient descent given in the previous section always leads to an isotropic nonlinear diffusion process without pre-smoothing ($\sigma = 0$ in (2.14)), it is often useful and sometimes inevitable to pre-smooth the gradient in the argument of the diffusivity g , see [54]. This has two effects: Firstly, it improves the stability properties of the process. For example, it has turned out that Perona-Malik diffusion is unstable without pre-smoothing, but is stabilised by pre-smoothing. Secondly, the removal of small-scale noise can be accelerated.

Presently, no energy functional is known that has (2.14) with $\sigma \neq 0$ as its gradient descent.

2.2.4 Special Singular Diffusivities

An important special case of (2.14) is constituted by the family of singular diffusivity functions [8, 73, 88, 116, 162, 197]

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|^p} \quad (p \geq 0) \quad (2.26)$$

which are unbounded and singular at zero. They include for $p = 1$ the well-known *total variation (TV)* diffusion, while for $p = 2$ the so-called *balanced forward-backward (BFB)* diffusion is obtained. Note that no pre-smoothing is involved here, so $\sigma = 0$.

2.2.4.1 Total Variation Diffusion

Setting $p = 1$ in (2.26) leads to the diffusivity

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|} \quad (2.27)$$

and thereby to the diffusion process [174]

$$\partial_t u = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (2.28)$$

which is known as *total variation diffusion* or *total variation flow (TV flow)* [6, 8, 73]. This flow arises also as gradient descent for the *total variation regularisation (TV regularisation)* functional [174, 1, 153]

$$E[u] = \int_{\Omega} |\nabla u| \, dx. \quad (2.29)$$

The latter also relates to the half-quadratic functionals [154, 181, 182] and Charbonnier functionals [62, 63], in which however the singularity is removed.

Total variation flow and total variation regularisation are frequently used in image processing and computer vision applications, either alone or as components of more complex dynamics or energy functionals. They have been the object of intensive theoretical investigation during the last 15 years [3, 7, 8, 21, 73]. Their favourable qualitative properties have also been confirmed by theoretical results, e.g., finite extinction time [7], shape preserving properties [21], edge preservation for these and other singular diffusivities [197].

The functional (2.29) displays the weakest growth that is possible for a convex penalisation of the gradient ∇u . Correspondingly, total variation flow is distinguished among isotropic diffusion processes as the one with the fastest decrease in diffusivity that still does not involve backward diffusion – note that we have constant flux $j(s) = 1$ and therefore $j'(s) = 0$, compare (2.19).

2.2.4.2 Balanced Forward-Backward Diffusion

Another interesting special case of (2.26) is obtained for $p = 2$. The corresponding diffusion process

$$\partial_t u = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|^2} \right) \quad (2.30)$$

is called *balanced forward-backward diffusion (BFB diffusion)* and has been studied by Keeling and Stollberger [116]. Unlike TV flow, it admits backward diffusion in regions of high gradients. It shares with TV flow the advantage to be free of parameters that need to be tuned to the image material, and the disadvantage of the singularity that makes it difficult to handle numerically.

To understand the name given to this process, we resort to the decomposition (2.18) which for BFB diffusion becomes

$$\partial_t u = \frac{1}{|\nabla u|^2} (\partial_{\xi\xi} u - \partial_{\eta\eta} u). \quad (2.31)$$

It is evident that the diffusion in gradient direction is always of backward type, and the intensity of this backward diffusion equals that of the forward diffusion along the level lines, establishing a “balance” between both contributions.

In dimensions other than 2, the representation (2.31) holds in modified form where $\partial_{\xi\xi}$ has to be replaced by a sum over all directions perpendicular to the local image gradient.

2.2.5 Multi-Channel Isotropic Nonlinear Diffusion

Similarly as in 2.1.3, we want to generalise the energy functional (2.20) to a multi-channel setting. Remembering what was said about the nature of Ψ in 2.2.2, one considers a penaliser function Ψ that determines how much the multi-channel gradient $(\nabla u_k)_{k \in \Gamma}$ encountered at some point contributes to the energy. Using the Euclidean norm in the vector space of multi-channel gradients, we arrive at the functional

$$E[u] = \frac{1}{2} \int_{\Omega} \Psi \left(\sum_{k \in \Gamma} |\nabla u_k|^2 \right) dx. \quad (2.32)$$

This formulation of the energy functional can be found, e.g., in [196] and for the total variation case in [31]. A calculation analogous to (2.22) shows that (2.22) possesses the gradient descent

$$\partial_t u_k = \operatorname{div} \left(\Psi' \left(\sum_{l \in \Gamma} |\nabla u_l|^2 \right) \nabla u_k \right). \quad (2.33)$$

Comparing this result to (2.21) shows that we have a nonlinear isotropic diffusion process in which all channels are diffused simultaneously using the same diffusivity $\Psi'(\cdot)$ that depends on data from all channels. This channel coupling for diffusion has been established in [92, 176, 204].

This is also plausible when the diffusion process is considered as a transport of information within the image domain: The *coupled diffusivity* makes sure that the data tuples $(u_k(x))_{k \in I}$ from different locations $x \in \Omega$ enter the transport process as a whole, and their averaging due to the diffusion process takes place in accordance with the vector space structure of \mathbb{R}^I .

The latter observation justifies to generalise (2.14) to the multi-channel setting in analogy to (2.33) as

$$\partial_t u_k = \operatorname{div} \left(g \left(\sum_{l \in I} |\nabla u_{l,\sigma}|^2 \right) \nabla u_k \right) \quad (2.34)$$

where $u_{l,\sigma} := K_\sigma * u_l$.

2.3 Anisotropic Nonlinear Diffusion

A further generalisation is obtained if the intensity of diffusion does not only depend on the location in the image domain but also on the direction. This bears considerable advantages e.g. for denoising images: While isotropic diffusion can only slow down the entire diffusion process near an edge, it is now possible to suppress diffusion *across* the edge, thus preserving it, while keeping upright a smoothing *along* the edge which still effectively reduces noise (edge-enhancing diffusion, [206]).

To achieve such an *anisotropic diffusion*, the scalar-valued diffusivity is replaced by a matrix-valued quantity, the *diffusion tensor*. The diffusion flow then depends on the gradient via a matrix-vector multiplication, such that its direction can differ from the gradient direction. The dependency of the diffusion tensor on the image structures is mediated by the structure tensor [91], which is basically a smoothed outer product tensor of the image gradients.

2.3.1 Basic Model

Anisotropic nonlinear diffusion. A class of anisotropic nonlinear diffusion equations is given by [206]

$$\partial_t u = \operatorname{div}(D(J) \cdot \nabla u) \quad (2.35)$$

where $D(J)$ is an anisotropic diffusion tensor which depends on the image via the *structure tensor* [91]

$$J = J_{\varrho}(\nabla u_{\sigma}) := K_{\varrho} * (\nabla(K_{\sigma} * u) \nabla(K_{\sigma} * u)^{\top}). \quad (2.36)$$

Here, K_{ϱ} and K_{σ} denote Gaussian convolution kernels. Initial and boundary conditions are chosen in analogy to the previously discussed cases.

By varying the parameters ϱ , σ and the way D depends on the structure tensor J , this equation can be adjusted to model a large class of anisotropic diffusion processes, including *edge-enhancing diffusion (EED)* [206] and *coherence-enhancing diffusion (CED)* [207].

2.3.1.1 Edge-Enhancing Diffusion

In this method, the diffusion tensor D has as its eigenvalues the gradient ∇u_{σ} of the pre-smoothed image and its orthogonal ∇u_{σ}^{\top} . The eigenvalue in direction ∇u_{σ} is set to $g(|\nabla u_{\sigma}|^2)$ where g is a nonincreasing nonnegative function with $g(0) = 1$, e.g. the Perona–Malik function $g(s^2) = (1 + s^2/\lambda^2)^{-1}$. Thereby the diffusion across edges is reduced depending on the contrast like in the isotropic diffusion model. The eigenvalue in direction ∇u_{σ}^{\top} is fixed to 1, ensuring that unattenuated diffusion takes place along an edge.

The diffusion tensor can therefore be expressed as

$$D(J_0) = D(\nabla u_{\sigma} \nabla u_{\sigma}^{\top}) = g\left(\nabla u_{\sigma} \nabla u_{\sigma}^{\top}\right), \quad (2.37)$$

i.e., dependent on the structure tensor for $\varrho = 0$. In this notation of D , we have made use of the fact that the application of a real-valued analytical function on a symmetric matrix is carried out by application to the eigenvalues, while leaving the eigenvectors unchanged.

2.3.1.2 Coherence-Enhancing Diffusion

This method [207] is designed to enhance line-like structures in images, which are characterised by strong coherence of local gradient directions without respect to orientation. While opposite gradients on both sides of a line in u cancel out when a larger σ is used to smooth ∇u_{σ} , they amplify each other if not ∇u itself is smoothed by Gaussian convolution but the outer product $\nabla u \nabla u^{\top}$. Therefore one computes here the structure tensor $J = J_{\varrho}(\nabla u_{\sigma} \nabla u_{\sigma}^{\top})$ with small σ (that just provides a moderate denoising and stabilisation of the direction information) and significantly larger ϱ (that determines the scale on which directional information is aggregated). Its eigenvector directions correspond to an averaged gradient orientation and its orthogonal,

while its anisotropy measured by difference or ratio of its eigenvalues is the higher, the stronger orientation is locally present in the image.

The diffusion tensor $D(J)$ inherits its eigenvector system from J . Its eigenvalue corresponding to the dominant (smoothed) gradient orientation is kept at a small fixed $\varepsilon > 0$ acting as a regularising “background diffusivity”, while diffusion perpendicular to gradients – thus, along detected lines – is intensified by raising the corresponding eigenvalue, the more anisotropic J is.

More specifically, if $J = \mu_1 e_1 e_1^T + \mu_2 e_2 e_2^T$ with $\mu_1 \geq \mu_2 \geq 0$ is the eigendecomposition of the structure tensor, one sets $D(J) := \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T$ with

$$\begin{aligned} \lambda_1 &:= \varepsilon, \\ \lambda_2 &:= \begin{cases} \varepsilon & \text{if } \mu_1 = \mu_2 \\ \varepsilon + (1 - \varepsilon) \exp\left(\frac{-C}{(\mu_1 - \mu_2)^2}\right) & \text{else,} \end{cases} \end{aligned} \quad (2.38)$$

where $C > 0$ is a contrast parameter.

2.3.2 Multi-Channel Anisotropic Nonlinear Diffusion

The extension of the anisotropic diffusion model to the case of multi-channel images follows the pattern from the isotropic model: The diffusion process in the individual channels is coupled by the use of one and the same diffusion tensor that depends on all channels simultaneously. As before, this ensures that the diffusion of image values $u(x) \in \mathbb{R}^F$ is consistent with the vector space structure of Γ . The multi-channel anisotropic diffusion model therefore reads [208]

$$\partial_t u_k = \operatorname{div}(D(J) \cdot \nabla u_k) \quad (2.39)$$

where the structure tensor of the multi-channel image $u = (u_k)_{k \in \Gamma}$ reads

$$J = J_\varrho(\nabla u_\sigma) := K_\varrho * \left(\sum_{k \in \Gamma} \nabla(K_\sigma * u_k) \nabla(K_\sigma * u_k)^T \right). \quad (2.40)$$

A difference compared to the single-channel case is that, due to the summation of the structure tensor over channels, even for $\sigma = 0$ and $\varrho = 0$ the eigenvector system of J is in general no longer aligned with the gradient directions ∇u_k of the individual channels. Therefore, true anisotropy is possible in (2.39) even without pre-smoothing.

2.3.3 Representation as Gradient Descent

Similar like its isotropic counterpart, anisotropic diffusion can be represented as a gradient descent of an energy functional if no smoothing operations are involved,

i.e., for $\sigma = 0$ and $\varrho = 0$.² However, such a process is not really anisotropic if a single-channel image is evolved; we therefore restrict our considerations here to the multi-channel case. We consider the energy functional

$$E[u] = \frac{1}{2} \int_{\Omega} \text{tr} \Psi \left(\sum_{k \in \Gamma} \nabla u_k \nabla u_k^T \right) dx \quad (2.41)$$

where Γ is again the index set of image channels, compare [211]. Here, Ψ is again understood as a decreasing function on \mathbb{R}_0^+ which is applied to the matrix argument $J_0(\nabla u) = \sum_{k \in \Gamma} \nabla u_k \nabla u_k^T$ in the canonical way, i.e., by applying it to each eigenvalue while retaining the eigenvector system of $J_0(\nabla u)$. This implies that if e_i denotes the i -th unit eigenvector of $J = J_0(\nabla u)$ (at some location), we have the identities

$$\text{tr} \Psi(J) = \sum_{i=1}^d e_i^T \Psi(J) e_i \quad (2.42)$$

and

$$e_i^T \Psi(J) e_i = \Psi(e_i^T J e_i). \quad (2.43)$$

Variational gradient. To verify our assertion on anisotropic diffusion as gradient descent, we turn now to calculate the variation of $E[u]$ as given in (2.41). In the derivation, we will make use of the identities (2.42) and (2.43). We substitute the vector-valued image u with $w = u + \varepsilon v$, where v is a vector-valued smooth function that vanishes on the domain boundary $\partial \Omega$. For component k we have then

$$\begin{aligned} \left. \frac{d}{d\varepsilon} E[w] \right|_{\varepsilon=0} &= \left. \frac{d}{d\varepsilon} \left(\frac{1}{2} \int_{\Omega} \text{tr} \Psi \left(\sum_{k \in \Gamma} \nabla w_k \nabla w_k^T \right) dx \right) \right|_{\varepsilon=0} \\ &= \frac{1}{2} \int_{\Omega} \left. \frac{d}{d\varepsilon} \left(\sum_{i=1}^d e_i^T \Psi \left(\sum_{k \in \Gamma} \nabla w_k \nabla w_k^T \right) e_i \right) \right|_{\varepsilon=0} dx \\ &= \frac{1}{2} \int_{\Omega} \left. \frac{d}{d\varepsilon} \left(\sum_{i=1}^d \Psi \left(\sum_k (\nabla w_k^T e_i)^T (\nabla w_k^T e_i) \right) \right) \right|_{\varepsilon=0} dx \\ &= \frac{1}{2} \int_{\Omega} \sum_{i=1}^d \left(\Psi' \left(\sum_{k \in \Gamma} (\partial_{e_i} u_k)^2 \right) \sum_{l \in \Gamma} \left. \frac{d}{d\varepsilon} \left((\partial_{e_i} w_k)^2 \right) \right|_{\varepsilon=0} \right) dx \\ &= \frac{1}{2} \int_{\Omega} \sum_{i=1}^d \left(\Psi' \left(\sum_{k \in \Gamma} (\partial_{e_i} u_k)^2 \right) \cdot \sum_{l \in \Gamma} \left. \frac{d}{d\varepsilon} \left((\partial_{e_i} u_k)^2 + 2\varepsilon \partial_{e_i} u_k \cdot \partial_{e_i} v_k + \varepsilon^2 (\partial_{e_i} v_k)^2 \right) \right|_{\varepsilon=0} \right) dx \end{aligned}$$

²No energy functional is known so far that has an anisotropic diffusion process of type (2.35) with $\sigma \neq 0$ or $\varrho \neq 0$ as gradient descent.

$$\begin{aligned}
&= \int_{\Omega} \sum_{i=1}^d \left(\Psi' \left(\sum_{k \in \Gamma} (\partial_{e_i} u_k)^2 \right) \sum_{l \in \Gamma} \partial_{e_i} u_l \cdot \partial_{e_i} v_l \right) dx \\
&= \sum_{l \in \Gamma} \sum_{i=1}^d \int_{\Omega} \Psi' \left(\sum_{k \in \Gamma} (\partial_{e_i} u_k)^2 \right) \partial_{e_i} u_l \cdot \partial_{e_i} v_l dx .
\end{aligned} \tag{2.44}$$

Integration by parts

$$\int_{\Omega} \Psi' \left(\sum_{k \in \Gamma} (\partial_{e_i} u_k)^2 \right) \partial_{e_i} u_l \cdot \partial_{e_i} v_l dx = - \int_{\Omega} \partial_{e_i} \left(\Psi' \left(\sum_{k \in \Gamma} (\partial_{e_i} u_k)^2 \right) \partial_{e_i} u_l \right) \cdot v_l dx \tag{2.45}$$

transfers this into

$$\left. \frac{d}{d\varepsilon} E[u + \varepsilon v] \right|_{\varepsilon=0} = - \int_{\Omega} \sum_{l \in \Gamma} \left(\operatorname{div} \left(\Psi' \left(\sum_{k \in \Gamma} \nabla u_k \nabla u_k^T \right) \nabla u_l \right) \cdot v_l \right) dx . \tag{2.46}$$

Remembering that the inner product in the function space that we are considering now reads

$$\langle u, v \rangle = \int_{\Omega} \left(\sum_{l \in \Gamma} u_l v_l \right) dx , \tag{2.47}$$

we drop integration, summation over image channels and multiplication by the factor v_l such that the sought variational derivative reads

$$\frac{\delta}{\delta u_l} E[u] = - \operatorname{div} \left(\Psi' \left(\sum_{k \in \Gamma} \nabla u_k \nabla u_k^T \right) \nabla u \right) . \tag{2.48}$$

Inserting this into (2.10) yields an anisotropic diffusion process of type (2.39) with $\sigma = 0$, $\varrho = 0$ and the diffusion tensor $D = \Psi'(J)$. Note that this also restricts the freedom in how the diffusion tensor may depend on the structure tensor: Each eigenvalue of D must depend exclusively on the corresponding eigenvalue of J , and the dependence must follow the same function Ψ' for all eigenvalues.

Part II

Multichannel M-Smoother

Chapter 3

M-Smoothers and Related Filters for Multi-Channel Images

Pollution of measured or computed multi-channel images with noise makes it a desideratum to provide an efficient and robust denoising filters for such data that do not destroy essential image features, like discontinuities (edges). A natural approach to this is to generalise existing filters for scalar-valued images.

Median filters lend themselves as a good choice because of their simplicity, efficiency and robustness. However, the straightforward approach to apply a scalar filter to the matrix components separately, which works fine for linear filters like Gaussian convolution, is not viable for non-linear filters like median filters.

Therefore we discuss here a matrix-valued median concept that takes its starting point from geometric and axiomatic considerations. A crucial point in this approach is to abandon rank orders which are inadequate for multi-channel data in favour of a minimality condition as defining property of medians. This concept has been introduced in the tensor image processing context in our work [218], and we have extended it in [217, 219, 225] to include mid-range filters and the more general class of M-smoothers with penaliser function $\psi(s) = s^p$, $p > 0$, as well as quantiles and weighted medians. Also, as alternatives to the originally considered Frobenius norm we included the spectral and nuclear norms. The present chapter integrates these contributions and partially follows the mentioned publications.

For the numerical computation of our matrix-valued filters, we present the gradient descent based algorithm proposed in [218]. As an attractive alternative to this, a convex optimisation framework has been presented in [217] and [225], which allows to take advantage of interior-point algorithms. This framework has been developed by F. Becker and C. Schnörr. I am also indebted to F. Becker for developing and

implementing experiments on smoothing structure tensors by median filtering which are presented in 3.1.9.4.

As already mentioned in [225], matrix-valued quantiles also relate to the matrix-valued morphological filters established in [47, 48] which rely on suitable supremum and infimum notions for sets of symmetric matrices.

Related work. Median filtering in signal processing has been established by Tukey [198] and has now become a standard technique in scalar-valued image processing, see Dougherty and Astola [78] or Klette and Zamperoni [122].

Denoising techniques for tensor data have been under intensive investigation recently, mainly in connection with DTI data, see the linear approaches by Westin et al. [229] or the nonlinear ones by Hahn et al. [102]. Nonlinear filters need to take into account the inherent relations of data matrices, either by channel coupling as in Tschumperlé and Deriche [195], or by working on derived quantities like eigendecompositions [72, 166, 195] or fractional anisotropy [159, 236].

Our matrix-valued median definition stands in the context of earlier attempts to vector-valued median filtering, see e.g. [17, 124]. In an image processing context, we mention Astola et al. [9] and Caselles et al. [53]. We stress that both definitions are built on the property of the median to be one of the *given* vectors, with a slight extension by admitting also their arithmetic mean in [9]. This property is also required in [18] by Barni et al. who otherwise use Euclidean distance sum minimisation similar to [218]. Surprisingly, for 2-D vectors already Austin's 1959 paper [10] proposes the exact analog of our definition given in [218]. Austin also gives a graphical algorithm which can be considered a direct predecessor of the gradient descent algorithm in [218]. Seymour's 1970 reply [183] to Austin discusses algorithmical difficulties and improvements of this procedure. Moreover, vector-valued medians and mid-range values (often by the name of 1-centres) have also been investigated in the literature on facility location problems, see the papers by Megiddo [140], Fekete et al. [87] and the references therein.

The convex optimisation numerics relies on concepts which can e.g. be found in the book by Boyd and Vandenberghe [34]. For applications in image processing contexts we mention Keuchel et al. [117].

The structure tensor has been established by Förstner and Gülch [91]. It is constructed by Gaussian smoothing of the outer product matrices $\nabla u \nabla u^T$ of the image gradient. To adapt better to orientation discontinuities, a modification called nonlinear structure tensor has been proposed by Weickert and Brox [41, 211]. Here, Gaussian smoothing is replaced by a nonlinear diffusion process. Nonlinear structure tensors have proven their use in texture segmentation [27, 41, 173] and motion analysis [43]. Another way to introduce structure adaptivity into the structure ten-

sor has been opened by van den Boomgaard and van der Weijer [199] who proposed a concept of robust structure tensor which is also linked to matrix-valued medians. For both adaptive structure tensor concepts, see also [42].

3.1 Median

3.1.1 Scalar-Valued Median

Let us first recall basic facts about scalar-valued median filtering.

Given a finite set of real numbers, their median is defined as the middle element in the sequence that contains these numbers ordered according to size. It can be considered as a robust average since it is insensitive to outliers in the given data set. The median operation commutes with monotone transformations of the data.

Without any reference to ordering, the median of the n -tuple $S = (a_1, \dots, a_n)$ can be characterised as the minimiser of the convex function

$$E_S(x) := \sum_{i=1}^n |x - a_i| \quad (3.1)$$

where $|x - a|$ is the Euclidean distance of real numbers.

If the number n of given data values is odd, this minimiser is uniquely defined; if n is even, all real numbers in the convex hull of the two middle data values (according to size) are minimisers with equal value. Often the convention is used to choose the arithmetic mean of the two values as median in this case; however, this introduces a metric notion that is actually foreign to the median concept. We adopt here the view that each minimiser can be considered as a median. At any rate, this detail is of little importance to our subsequent considerations.

The median concept gives rise to a local image filter with interesting properties. *Median filtering* requires the specification of a neighbourhood for each pixel which is commonly chosen either as a $(2k + 1) \times (2k + 1)$ square or a discretely approximated disc centred at the pixel. The new grey-value of a pixel is obtained as the median of the old grey-values within the neighbourhood.

Median filtering can be iterated and so constitutes a discontinuity-preserving denoising process. The insensitivity of medians to outliers enables median filtering to cope even with extreme types of noise like uniform or impulse noise.

Unlike Gaussian smoothing, median filtering possesses non-constant steady states called *root signals*. Although for discrete median filtering not each initial image converges to a root signal – a counter-example is given by an image made up of alternating

black and white rows of pixels – the ability of iterated median filtering to preserve discontinuities is one of its most interesting properties. For a more detailed discussion of root signals, see e.g. [82].

For a space-continuous variant of median filtering, Guichard and Morel [99] have proven that it approximates mean curvature flow [3, 4, 118], thereby establishing a remarkable link between a nonlinear local filter and a PDE-based image evolution. Guichard and Morel’s result actually covers a larger filter class. We recall it here only restricted to the median case:

Let $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ be a continuous function of bounded variation, and let B_ϱ be a closed disk of radius ϱ for $\varrho > 0$. Let the function u be obtained by median filtering of f with structure element B_ϱ as follows: For each $(x, y) \in \mathbb{R}^2$ let $u(x, y)$ be the 0.5-quantile of the (continuous) histogram of

$$f((x, y) + B_\varrho) := \{f(\xi, \eta) \mid (\xi - x, \eta - y) \in B_\varrho\}. \quad (3.2)$$

Then one has

$$\lim_{\varrho \rightarrow +0} \frac{u(x, y) - f(x, y)}{\varrho^2/6} = |\nabla f| \operatorname{div} \left(\frac{\nabla f}{|\nabla f|} \right). \quad (3.3)$$

Remark. This means that one median filtering step with structure element B_ϱ approximates an (explicit forward) time step of the partial differential equation of (mean) curvature motion

$$\partial_t u = |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (3.4)$$

with initial condition $u(t=0) = f$ and time step size $\tau = \varrho^2/6$.

3.1.2 Multi-Channel Median Definition

We turn now to generalising median filtering to matrix-valued images. The main task in doing so is to give an appropriate notion of medians for matrices since the construction of the local image filter by applying the median to input values from a neighbourhood transfers straightforward.

While not all properties of scalar-valued medians can be retained by such a generalisation, the following requirements are essential from the modeling viewpoint:

Preservation of symmetry. The median of symmetric matrices must again be a symmetric matrix.

Scaling invariance. For a real number λ , the median med should satisfy

$$\operatorname{med}(\lambda A_1, \dots, \lambda A_n) = \lambda \operatorname{med}(A_1, \dots, A_n) \quad (3.5)$$

for arbitrary input matrices A_1, \dots, A_n .

Rotational invariance. Rotating all input matrices by the same rotation matrix R should result in equal rotation of the median:

$$\text{med}(R^T A_1 R, \dots, R^T A_n R) = R^T \text{med}(A_1, \dots, A_n) R. \quad (3.6)$$

Embedding of scalar-valued median. If all input matrices are scalar multiples of the same non-zero matrix A , the median should reduce to the scalar median:

$$\text{med}(\lambda_1 A, \dots, \lambda_n A) = \text{med}(\lambda_1, \dots, \lambda_n) A. \quad (3.7)$$

Preservation of positive semidefiniteness. Since positive semidefiniteness is an indispensable property of some sorts of matrix data, such as DTI or structure tensor fields, a sensible filter for such data should not destroy it.

Remark. If the matrix data being processed represent a tensor field, the rotational invariance condition given here appears more restrictive than necessary in the light of our discussion in 1.4 since it applies to rotations of the matrix data independent on transformations of the underlying image domain. This is a consequence of the median filter setting which selects the data values from some neighbourhood and disregards their spatial arrangement within that neighbourhood. A less restrictive condition referring only to physically meaningful rotations of the tensor field – such that rotate values in accordance with the transformation of the underlying image domain – would require to modify the neighbourhood filter concept in a suitable way. While we do not study such modifications in the present work, they clearly constitute an interesting direction for further research.

The constraints of positive semidefiniteness preservation and rotational invariance narrow drastically the range of possible definitions. In particular, the naive idea of taking the scalar median in each matrix component is ruled out easily: Let the three positive definite matrices

$$\begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.8)$$

be given. Their component-wise median

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \quad (3.9)$$

is indefinite. By a rotation of the given matrices, one obtains

$$\begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix}, \quad \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad (3.10)$$

whose component-wise median

$$\begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.11)$$

is positive semidefinite. We see that neither rotational invariance nor preservation of positive semidefiniteness is achieved in this way.

Further, since matrices lack a linear ordering, a rank-order approach to defining matrix medians is impractical. Instead, we generalise the minimising property (3.1).

Definition 3.1 *Given a tuple $S = (A_1, \dots, A_n)$ of matrices of equal size, the minimiser of*

$$E_S(X) = \sum_{i=1}^n \|X - A_i\|, \quad (3.12)$$

where $\|\cdot\|$ is a matrix norm, is called median of S and denoted by $\text{med}(S)$.

Remark. The definition can easily be generalised even further by replacing $\|X - A_i\|$ with some metric $d(X, A_i)$. We refrain from further considerations in this generality and stick with metrics derived from norms.

Clearly, E_S is convex for any norm $\|\cdot\|$. In some cases it may not be strictly convex, such that besides the desirable situation of a unique minimiser also convex closed sets of minimisers can occur.

However, rotational invariance and semidefiniteness preservation restrict the choice of $\|\cdot\|$ more radically. Before discussing possibilities, let us notice one property of the median which is independent of the norm.

Lemma 3.1 *Let $X = \text{med}(A_1, \dots, A_n)$. If each A_i is replaced by $A'_i := X + k_i(A_i - X)$ with real $k_i > 0$, then X is also the median of A'_1, \dots, A'_n .*

Accordingly, the matrices A_i can be shifted along the rays from X to A_i without affecting the median. The statement follows directly from the scaling property $\|kA\| = |k|\|A\|$, $k \in \mathbb{R}$. It can be considered a restricted form of the independence on outliers known from the scalar-valued median.

3.1.3 Choice of Norms

In the following, we discuss one possible norm for the case of vector fields and three norms for $d \times d$ -matrices. All of the latter are constructed from the eigenvalues $\lambda_1(A), \dots, \lambda_d(A)$ of A , which guarantees rotational invariance.

3.1.3.1 Vector-Valued Case: Euclidean Norm

For vector fields, the Euclidean norm

$$\|v\|_{(2)} = \sqrt{\sum_{j=1}^d v_j^2} \quad (3.13)$$

enjoys clear preference for its rotational invariance.

For our later investigations, the convexity properties of the Euclidean norm are important. Firstly, $\|\cdot\|_{(2)}$ is a convex function in \mathbb{R}^d , i.e.,

$$\|\alpha v_1 + (1-\alpha)v_2\|_{(2)} \leq \alpha \|v_1\|_{(2)} + (1-\alpha)\|v_2\|_{(2)} \quad (3.14)$$

holds for $v_1, v_2 \in \mathbb{R}^d$ and $0 \leq \alpha \leq 1$.

Secondly, $\|\cdot\|_{(2)}$ is strictly convex along every straight line except on half-lines starting at the origin. This means that besides the trivial cases $v_1 = v_2$ or $\alpha = 0$ or $\alpha = 1$ equality holds in (3.14) only if v_1 and v_2 are nonnegative multiples of each other.

We remark that the convexity of $\|\cdot\|_{(2)}$ as well as the non-strictness along half-lines starting at the origin are consequences of the defining properties of a norm.

3.1.3.2 Matrix-Valued Case: Frobenius Norm

As the matrix analogue of the Euclidean norm, we consider the Frobenius norm which can also be computed directly from the matrix entries a_{jk} , $j, k = 1, \dots, d$, of A ,

$$\|A\|_{(2)} = \sqrt{\sum_{j=1}^d |\lambda_j(A)|^2} = \sqrt{\sum_{j,k=1}^d a_{jk}^2}. \quad (3.15)$$

With respect to convexity, all that has been said about the Euclidean norm holds equally for the Frobenius norm.

3.1.3.3 Matrix-Valued Case: Spectral Norm

Second, we have the spectral norm

$$\|A\|_{(\infty)} = \max_{j=1, \dots, d} |\lambda_j(A)|. \quad (3.16)$$

It is the operator norm with respect to the action of matrices as operators on vector spaces. Its computation is comparably expensive. One possibility is an eigendecomposition; however, since only the eigenvalue with largest modulus is relevant, there

are cheaper alternatives like van Mises' algorithm that relies on iterated matrix-vector multiplications.

For the spectral norm, non-strict convexity occurs not only in those cases which are required by the norm properties and which are observed for the Euclidean or Frobenius norm: Consider a fixed eigenvector system in \mathbb{R}^d . The space of symmetric $d \times d$ matrices with this eigenvector system then decomposes into domains in each of which the spectral norm is linear, thus non-strictly convex. Since this linearity within one domain in fact only depends on the equal orientation of one eigenvector (that for the largest eigenvalue), it is easy to give even hyperplane segments, i.e., domains with codimension 1 in the space of symmetric $d \times d$ matrices, within which linearity holds.

3.1.3.4 Matrix-Valued Case: Nuclear Norm

Our last norm for matrices is the so-called *nuclear norm* which is given by

$$\|A\|_{(1)} = \sum_{j=1}^d |\lambda_j(A)| \quad (3.17)$$

For positive semidefinite matrices, we have $\|A\|_{(1)} = \text{tr}(A)$, which allows an efficient computation in this case. Typically, however, even when the A_i are positive semidefinite, the differences $X - A_i$ aren't.

We remark that the Frobenius and nuclear norms are examples of the family of norms

$$\|A\|_{(p)} = \left(\sum_{j=1}^d |\lambda_j(A)|^p \right)^{1/p}, \quad p \geq 1 \quad (3.18)$$

(namely, $p = 1$ for the nuclear and $p = 2$ for the Frobenius norm), which also includes the spectral norm as limit case $p \rightarrow \infty$.

The nuclear norm is linear and thus non-strictly convex in large domains of the space of symmetric matrices, one of them being the entire cone of positive semidefinite symmetric matrices (since it equals the trace there).

3.1.4 Frobenius Median

In the following, we study the properties of multi-channel medians in more detail for the case of matrix-valued data with the three norms enumerated in Section 3.1.3. For brevity, we shall refer to medians defined via these norms as *Frobenius median* med_2 , *spectral median* med_∞ , and *nuclear median* med_1 .

We start our discussion of the Frobenius median with a remark on uniqueness. Since the Frobenius norm is strictly convex along almost all lines (see 3.1.3), the energy $E_S(X)$ in the definition of the Frobenius median is strictly convex everywhere except if all given data A_1, \dots, A_n lie along one straight line. In this case, the situation is the same as for the scalar median. Otherwise there is always a unique minimiser.

Another useful observation is given in the following proposition.

Proposition 3.2 *The Frobenius median $\text{med}_2(S)$ of a tuple $S = (A_1, \dots, A_n)$ of $d \times d$ matrices is a convex combination of A_1, \dots, A_n .*

Proof. The Frobenius norm coincides with the Euclidean norm if the $d \times d$ matrices are interpreted as vectors in \mathbb{R}^{d^2} . We identify therefore matrices with vectors and denote by $\langle \cdot, \cdot \rangle$ the corresponding Euclidean scalar product. Assume now that X is a matrix outside the convex hull of A_1, \dots, A_n . Then a hyperplane h separates X from A_1, \dots, A_n . Let Y be the orthogonal projection of X onto h , i.e., $X - Y$ is perpendicular to h . Then $\langle X - Y, Y - A_i \rangle$ is positive for $i = 1, \dots, n$. Hence,

$$\begin{aligned} & \langle X - A_i, X - A_i \rangle - \langle Y - A_i, Y - A_i \rangle \\ &= \langle X, X \rangle - 2\langle X, A_i \rangle - \langle Y, Y \rangle + 2\langle Y, A_i \rangle \\ &= 2\langle X - Y, Y - A_i \rangle + \langle X - Y, X - Y \rangle > 0 \end{aligned} \tag{3.19}$$

which proves that X is not the minimiser of (3.12). \square

Since convex combinations of positive semidefinite matrices are positive semidefinite, the following corollary is obvious.

Corollary 3.3 *The Frobenius median $\text{med}_2(S)$ of a tuple $S = (A_1, \dots, A_n)$ of positive semidefinite symmetric $d \times d$ matrices is positive semidefinite.*

The matrix–vector identification used in the proof of Proposition 3.2 shows that this median definition is not restricted to square matrices but works equally on non-square matrices, including vectors, for which it seems more adequate to speak of *Euclidean median*. For \mathbb{R}^2 , Austin’s bivariate median [10] is recovered.

This simple planar Euclidean case of 2-dimensional vectors allows us to illustrate simple geometric properties of our median concept. Three points in the plane which span a triangle with all angles smaller than 120 degrees have as their median the so-called Fermat–Torricelli or Steiner point. From this point, each connecting line between two of the given points appears under a 120 degree angle (see Figure 3.1(a)). If instead one angle of the triangle is larger or equal 120 degrees, then its vertex is the median (see Figure 3.1(b)). In the case of four points spanning a convex quadrangle,

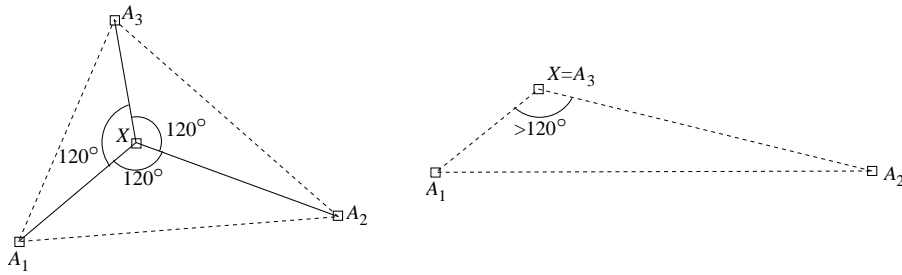


Figure 3.1: Median X of three non-collinear points A_1, A_2, A_3 in the Euclidean plane. *Left: (a)* If no interior angle of the triangle $A_1A_2A_3$ is greater than 120° , then X is the Fermat–Torricelli point, or Steiner point. This is the unique point from which each side of $A_1A_2A_3$ appears under a 120° angle. *Right: (b)* If the triangle $A_1A_2A_3$ has an interior angle greater than 120° , then the vertex of this angle is the median (here, $X = A_3$).

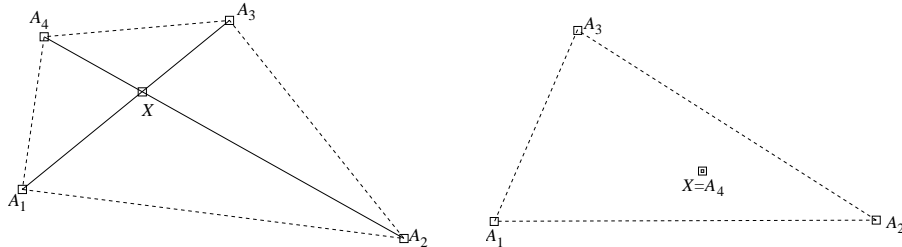


Figure 3.2: Median X of four points A_1, A_2, A_3, A_4 (no three collinear) in the Euclidean plane. *Left: (a)* If the four points span a convex quadrangle (here, $A_1A_2A_3A_4$), then X is the intersection points of the diagonals (here, A_1A_3 and A_2A_4). *Right: (b)* If the convex hull of the four points is a triangle (here, $A_1A_2A_3$), then the one point which is not a vertex of this triangle is the median (here, $X = A_4$).

the median is the intersection point of the diagonals (Figure 3.2(a)). The median of four points whose convex hull is a triangle is the one of the points which is not a corner of this triangle (Figure 3.2(b)).

Combinatorial and geometric complexity prevents similar elementary geometric considerations for more points or higher dimensions. It is evident, though, that although our definition does not force the median to be one of the given data points, this still happens to be true in many generic cases. Only if none of the given values is located sufficiently well in the middle of the data set, a new value is created.

It can therefore be expected that multi-channel median filters behave similarly robust with respect to outliers as their scalar-valued counterparts, and that even root signals arise since after some iterations of the median filter the local distribution of values is changed such that almost always one of the given values is selected as median. Our experimental results presented later confirm this expectation.

3.1.5 Nuclear Median

With the nuclear norm, the energy (3.12) displays non-strict convexity in a broader range of configurations, leading to non-unique minimisers. Our result on semidefiniteness is therefore weaker than before.

Proposition 3.4 *Let a tuple $S = (A_1, \dots, A_n)$ of positive semidefinite $d \times d$ matrices be given, and consider the objective function E_S with the nuclear norm. If E_S is minimised by a matrix which is not positive semidefinite, then there exists also a positive semidefinite argument for which E_S attains the same value.*

Proof. We consider a symmetric matrix X whose smallest eigenvalue μ is negative. The difference matrix $X - A_i$ for any A_i has two eigenvalues $\lambda_1 \geq \lambda_2$ where $\lambda_2 \leq \mu$. The matrix $X - \mu I - A_i$ has the same eigensystem as $X - A_i$, with both eigenvalues shifted by μ . From $\lambda_2 \leq \mu$ it follows that

$$\begin{aligned} \|X - \mu I - A_i\|_{(1)} &= |\lambda_1 - \mu| + |\lambda_2 - \mu| \\ &\leq |\lambda_1| + |\mu| + |\lambda_2| - |\mu| = \|X - A_i\|_{(1)}. \end{aligned} \tag{3.20}$$

This proves the statement of the proposition. \square

As another remarkable property of the nuclear median, we mention that it reveals an insensitivity w.r.t. outliers which goes beyond the one described in Lemma 3.1 and is in fact close to the corresponding property of its scalar-valued counterpart.

Lemma 3.5 *Let $X = \text{med}_1(A_1, \dots, A_n)$. Assume that for the data matrix A_i , the difference $X - A_i$ is positive or negative definite. If A_i is replaced by some other A'_i for which $X - A'_i$ has the same (positive or negative) definiteness as $X - A_i$, then X is also the nuclear median of $A_1, \dots, A'_i, \dots, A_n$.*

A consequence of this behaviour is that the orientation of the median X depends exclusively on those A_i for which $X - A_i$ is indefinite. Since this can lead to orientations of the nuclear median matrix that do not represent well the orientation set of the given data, this feature put nuclear median filtering at some disadvantage as compared to the Frobenius alternative.

3.1.6 Spectral Median

Though the convexity of the spectral norm is non-strict in domains of hyperplanes, the energy E_S in the spectral median definition is strictly convex for sufficiently large

generic data sets S since the principal eigenvector directions of $X - A_i$ will often not coincide for all i . So, if non-uniqueness takes place for the spectral median, the sets of minimisers are smaller than in case of the nuclear median.

We describe in detail the situation for symmetric 2×2 matrices.

Lemma 3.6 *Let n symmetric 2×2 -matrices A_1, \dots, A_n be given whose spectral median is X . Assume that $A_i = \begin{pmatrix} a_i & c_i \\ c_i & b_i \end{pmatrix}$, $i = 1, \dots, n$, and $X = \begin{pmatrix} x & z \\ z & y \end{pmatrix}$. Then the following are true:*

- *The trace of X is a scalar median of the traces of A_1, \dots, A_n (in the sense that it minimises (3.1)).*
- *The vector $(x - y, 2z)^T$ is the bivariate median w.r.t. Euclidean norm of the vectors $(a_i - b_i, 2c_i)$, $i = 1, \dots, n$.*

Proof. We calculate the energy expression $E_S(X) = \sum_i \|X - A_i\|_{(\infty)}$ explicitly. The eigenvalues of $X - A_i$ are given by

$$\lambda_{\pm} = \frac{1}{2} \left(x - a_i + y - b_i \pm \sqrt{(x - a_i - y + b_i)^2 + 4(z - c_i)^2} \right). \quad (3.21)$$

Clearly, we have $|\lambda_+| \geq |\lambda_-|$ if $x - a_i + y - b_i \geq 0$, and vice versa, thus

$$\begin{aligned} \|X - A_i\|_{(\infty)} = \frac{1}{2} \left(\sqrt{((x - y) - (a_i - b_i))^2 + (2z - 2c_i)^2} \right. \\ \left. + ((x + y) - (a_i + b_i)) \operatorname{sgn}((x + y) - (a_i + b_i)) \right) \end{aligned} \quad (3.22)$$

and finally

$$\begin{aligned} \sum_i \|X - A_i\|_{(\infty)} = \frac{1}{2} \sum_{i=1}^n \|(x - y, 2z)^T - (a_i - b_i, 2c_i)^T\|_{(2)} \\ + \frac{1}{2} \sum_{i=1}^n |\operatorname{tr} X - \operatorname{tr} A_i|. \end{aligned} \quad (3.23)$$

The two sums on the right-hand side can be minimised separately, which gives us exactly the two conditions stated in the Lemma. \square

We see that in the case of n odd (which is of practical relevance as it occurs in any centrally symmetric discrete structure element) the spectral median is always uniquely determined.

Unfortunately, the spectral median deviates from the previously discussed matrix-valued median concepts in that it does not always preserve positive semidefiniteness.

To demonstrate this, we use Lemma 3.6 to compute the (uniquely determined) spectral median of the three positive definite 2×2 matrices

$$\begin{pmatrix} 2.95 & 0.42 \\ 0.42 & 0.07 \end{pmatrix}, \begin{pmatrix} 2.95 & -0.42 \\ -0.42 & 0.07 \end{pmatrix}, \begin{pmatrix} 4 & 0 \\ 0 & 0.01 \end{pmatrix}. \quad (3.24)$$

Their traces are 3.02, 3.02, 4.01 such that 3.02 is also the trace of the sought spectral median. Furthermore, the three points $(2.88, 0.84)^T, (2.00, -0.84)^T, (3.99, 0.00)^T$ form an isosceles planar triangle. Its Steiner point lies on the symmetry median (the first coordinate axis); the line from the Steiner point to $(2.88, 0.84)^T$ encloses a 120° angle with the positive symmetry axis. Thus, the Steiner point is $(2.88 + 0.84/\sqrt{3}, 0)^T$ leading to the median

$$\begin{pmatrix} 2.95 + 0.14\sqrt{3} & 0 \\ 0 & 0.07 - 0.14\sqrt{3} \end{pmatrix} \approx \begin{pmatrix} 3.2425 & 0 \\ 0 & -0.1925 \end{pmatrix} \quad (3.25)$$

which is indefinite.

3.1.7 Numerics: Gradient Descent with Step Size Adaptation

Only in simple cases it is possible to compute matrix medians directly. In general numerical approximation methods are required.

The Fermat Problem, i.e., the Euclidean median of \mathbb{R}^d -valued data, has been considered among others by Weiszfeld [216] and Austin [10]. Both authors also proposed algorithms: Weiszfeld's algorithm is an iterative numerical computation suitable for arbitrary dimension d while Austin formulated a graphical method for $d = 2$ which can be translated into a sort of gradient descent. Both algorithms encounter difficulties: Kuhn [126] pointed out that Weiszfeld's iteration can be trapped in non-minima, while Seymour [183] criticised problems in Austin's method related to the determination of a proper step size.

The algorithm that we will discuss now is related to both previously presented methods, and it underlies most of the experiments on matrix-valued medians presented in this chapter. Though in the case of Frobenius medians it may be less efficient than e.g. Weiszfeld's method, it enjoys the advantage of greater generality since its strictly local approach adapts easily to different norms or metrics.

3.1.7.1 Frobenius Median

In computing Frobenius medians of matrices (or Euclidean medians of vector data), the convexity of $E_S(X)$ and its differentiability except at $X = A_i$ motivate the use of gradient descent techniques. One difficulty has to be overcome: Since the gradient

vector $\nabla \|A_i - X\|_{(2)}$ has equal length for all $X \neq A_i$, it lacks any information about the distance to A_i . This deficiency is inherited by the gradient of E_S . Though clearly indicating the *direction* for descent, it is useless in determining how far to go in one step. A remedy for this is to use an adaptive step-size control which uses information from the over- and undershoots encountered during iteration.

For simplicity of notation, let us define $\nabla \|X - A\|_{(2)} := 0$ if $X = A$, such that $\nabla E_S(X)$ is defined everywhere. Our algorithm in its basic form then reads as follows.

1. Find the index $j \in \{1, \dots, n\}$ for which $E_S(A_j)$ is minimal. If the inequality $\left\| \nabla \sum_{i \neq j} \|X - A_i\|_{(2)} \right\| \leq 1$ holds for $X = A_j$, then A_j is also the global minimiser – stop. Otherwise proceed to step 2.
2. Let $X_0 := A_j$, $k := 0$ and choose an arbitrary initial step size $s_0 > 0$.
3. Compute the gradient $\nabla E_S(X_k)$. If the norm of this gradient is below a given threshold $\varepsilon > 0$, consider X_k as approximate median – stop.
4. Perform one step of gradient descent

$$X_{k+1} := X_k - s_k \nabla E_S(X_k). \quad (3.26)$$

5. To detect over- and undercorrections and to adapt the step size, compute the indicator

$$r := \frac{\langle \nabla E_S(X_{k-1}), \nabla E_S(X_k) \rangle}{\langle \nabla E_S(X_k), \nabla E_S(X_k) \rangle} \quad (3.27)$$

which measures the projection of $\nabla E_S(X_k)$ onto $\nabla E_S(X_{k-1})$. If $r < 0$, we assume that an overshoot has occurred, while $r \gg 0$ signals an undershoot.

6. Let $s_{k+1} = s_k / (1 - r)$.
7. If $r < r_{\text{crit}}$ with a fixed $r_{\text{crit}} \in (-1, 0)$, let $s_k := s_{k+1}$ and go back to step 4 (i.e., roll back the last iteration step, and repeat it with reduced step size).
8. Increase k by one, and proceed with step 3.

In practice, it is useful to refine the update rule for the step size in step 6 of the algorithm e.g. by projecting r onto the interval $[-1/2, 2]$ first in order to avoid large jumps in the step size.

3.1.7.2 Spectral Median

While the Frobenius norm is differentiable everywhere except at zero, the spectral norm displays additional singularities along the hypersurfaces of matrices with multiple eigenvalues. The energy expression $E_S(X)$ for the spectral median inherits these

hypersurface-singularities which arise from the maximum operation applied on two differentiable functions $f_1(X)$ and $f_2(X)$, namely the absolute values of different eigenvalues.

We use therefore an approximation to $E_\zeta(X)$ which is differentiable everywhere outside the A_i . It is obtained by replacing $\max(f_1(X), f_2(X))$ with $wf_1(X) + (1-w)f_2(X)$ where $w = w(f_1(X) - f_2(X))$ is a smoothed Heaviside function. The function w can for instance be generated by convolution of the Heaviside function H with a Gaussian K_ε or a B-spline B_ε^k of order¹ $k \geq 2$ with support $[-\varepsilon, \varepsilon]$, i.e., $w = K_\varepsilon * H$ or $w(s) = B_\varepsilon * H$. The gradient descent algorithm then works as before.

3.1.7.3 Nuclear Median

Similarly, the nuclear norm is non-differentiable at singular matrices. This time, hypersurface-singularities arise directly from taking the absolute values of eigenvalues. Denoting by $f_1(X)$ and $f_2(X)$ an eigenvalue and its negative, the same smoothing procedure as for the spectral median can be applied to make the gradient descent algorithm applicable.

3.1.8 Numerics: Convex Optimisation

Another attractive method to compute matrix medians which bypasses elegantly the difficulties of the gradient descent starts directly from the optimisation form of our definition. By a chain of transformations, the median definition is translated into a convex optimisation problem that admits the use of established and efficient interior point algorithms. This approach has been contributed mainly by the work of F. Becker and C. Schnörr, and it is documented in [20, 217, 225].

¹Here, we use the notation $B_\varepsilon^k = \underbrace{B_{\varepsilon/(k+1)}^0 * \dots * B_{\varepsilon/(k+1)}^0}_{k+1 \text{ convolution factors}}$ where B_δ^0 is a unit-weight box function supported on $[-\delta, \delta]$. Clearly, B_ε^k is $(k-1)$ times continuously differentiable.

3.1.9 Experimental Evaluation

3.1.9.1 Filtering Synthetic Matrix Data

To demonstrate basic features of matrix-valued median filtering, we start by a simple experiment on a synthetic image consisting of symmetric positive definite 2×2 matrices, see Figure 3.3.

The data set in Figure 3.3(a) contains one discontinuity along a diagonal line while the values vary smoothly elsewhere. In Figure 3.3(b) it has been filtered by a few Frobenius median iterations with reflecting boundary conditions. It can be seen that the discontinuity is nicely preserved. Close to the image boundaries, the discontinuity is slightly displaced due to the reflecting boundary conditions. They let the discontinuity appear as a corner in those regions, which is rounded off such that the discontinuity is bent towards the boundary normal.

In Figure 3.3(c) the eigenvalues of the matrices from (a) have been changed by adding Gaussian noise (with cutoff at zero and an upper bound) while the eigenvector directions have been retained. The median filtered image (d) demonstrates the effective denoising as it differs from (b) only by a moderate decay in anisotropy of the matrices (such that the ellipses appear rounder), while the discontinuity is clearly preserved.

3.1.9.2 Smoothing DT-MRI Data

In Figures 3.4–3.7 results of median filtering of DT-MRI data are shown. For visualisation purposes and because the median filter behaves fully analogous in higher dimensions, we demonstrate again the filtering of 2×2 matrices in a planar image domain. The original data set shown in Figure 3.4(a) consists of the upper left four components of the transaxial brain section shown in Figure 1.4(a), which correspond to the diffusivity components within the transaxial plane.

To investigate the denoising capabilities of median filtering, matrix-valued impulse noise has been added in Figure 3.4(b) with a ratio of 30% noise pixels. All median filtering procedures have been applied to this noisy image.

The further frames of Figure 3.4 show results of Frobenius median filtering with 3×3 stencil and increasing iteration counts. After about 30 iterations, there are almost no further changes – the filter has reached practically a root signal. Figure 3.5 shows a detail from the corpus callosum region of the original image, noisy image and the first two filtering results in ellipse visualisation.

Figure 3.6 illustrates the effect of increasing stencil size with constant iteration count (3 iterations). Filtering results with more iterations of larger stencils have not been

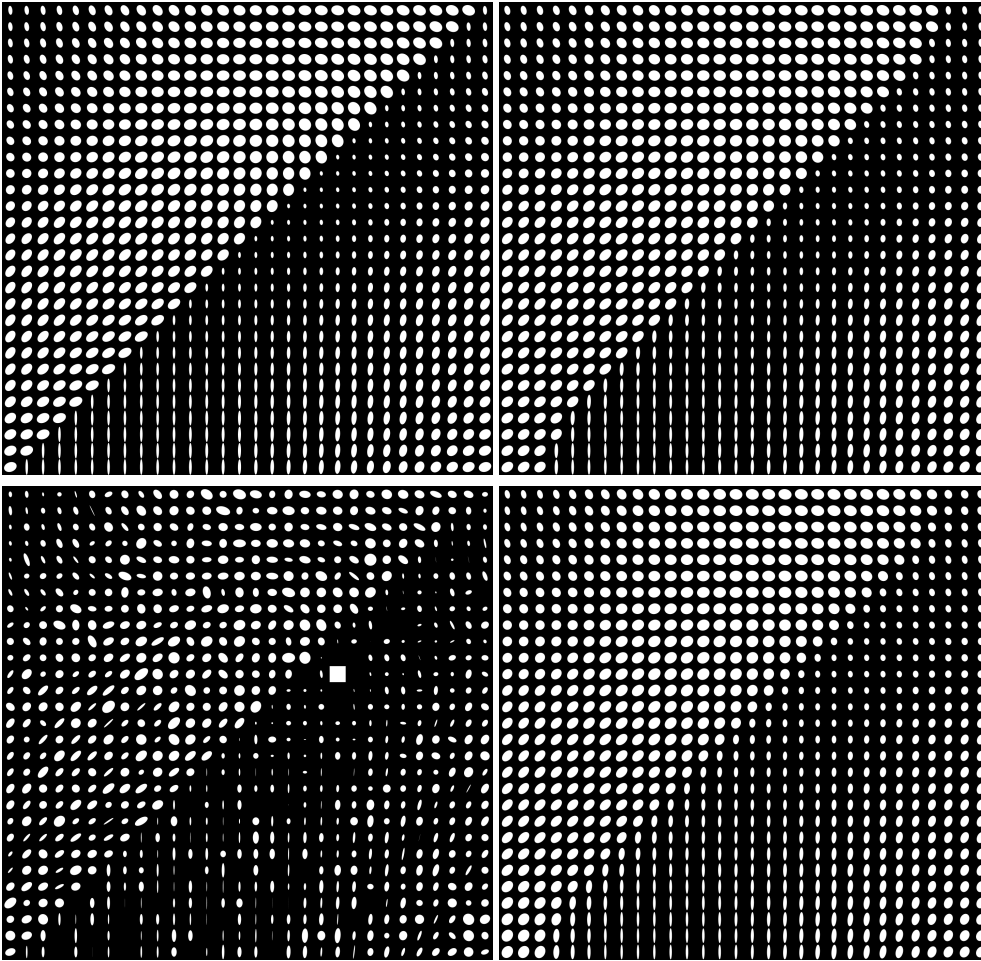


Figure 3.3: Discontinuity-preserving filtering of matrix data. *Top left: (a)* A field of positive definite matrices with a discontinuity, visualised by ellipses. *Top right: (b)* Filtered by 5 iterations of Frobenius median with 5×5 stencil. *Bottom left: (c)* Image (a) degraded by Gaussian noise applied to the eigenvalues. The eigenvector directions are unchanged. The square represents one matrix whose eigenvalues are too large to be displayed. *Bottom right: (d)* Image (c) filtered by Frobenius median as in (b).

included since at the moderate image resolutions of the DTMRI data e.g. a 7×7 stencil is already way too large for most of the interesting image features, leading to constant or almost constant images after some tens of iterations.

Finally, we contrast in Figure 3.7 median filters based on the three different norms discussed before. It is evident that in spite of their theoretical differences, the three variants show hardly any visible distinction on our real-world data.

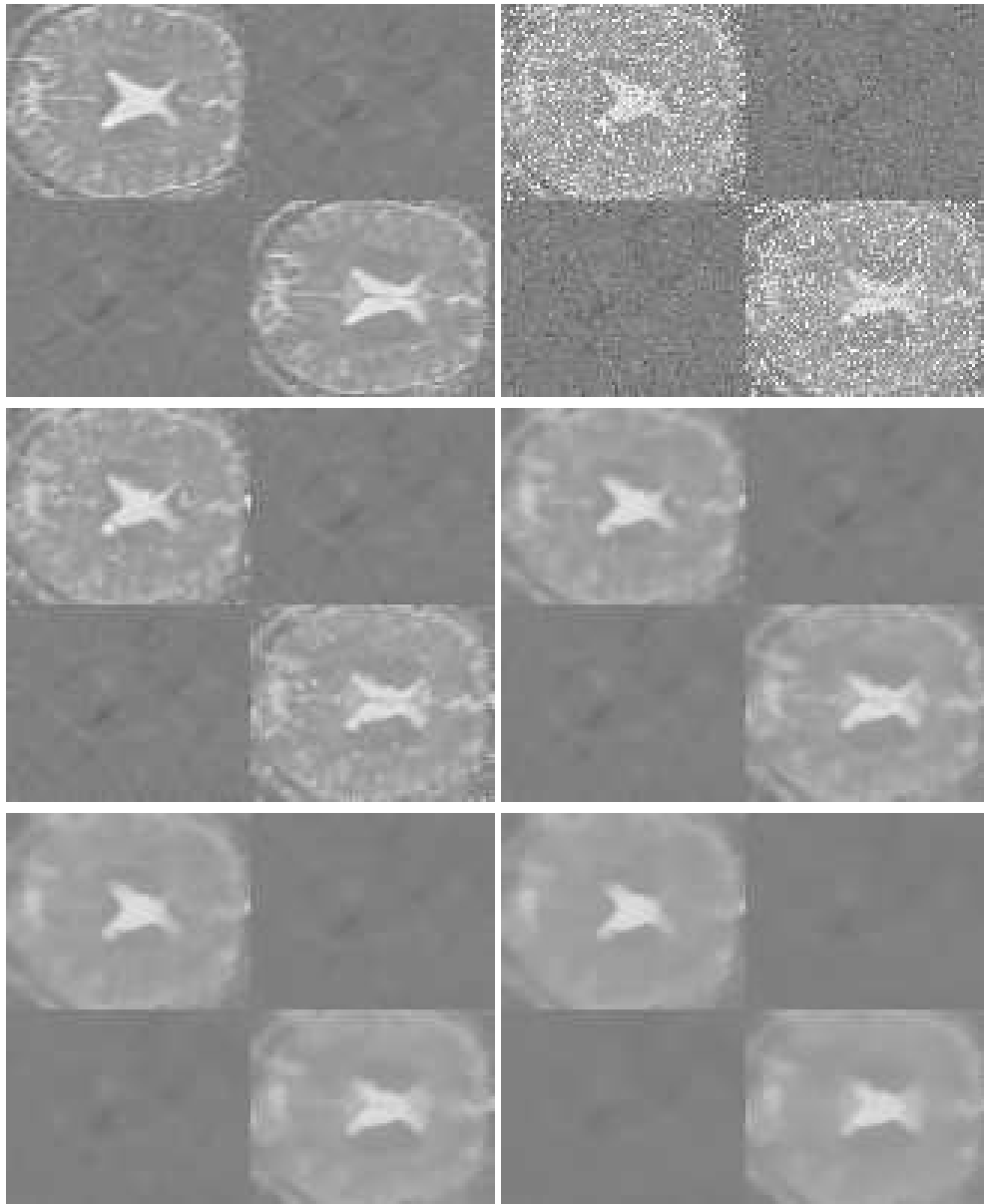


Figure 3.4: Frobenius median filtering of 2D DTMRI data. *Top left: (a)* One slice from a DTMRI brain scan. Only tensor components belonging to the cut plane are shown. *Top right: (b)* 30% of the matrices of (a) have been replaced with uniform noise (uniform in directions and uniform in both eigenvalues). *Middle left: (c)* Image (b) filtered by Frobenius median, 3×3 stencil, 1 iteration. *Middle right: (d)* Frobenius median, 3×3 stencil, 3 iterations. *Bottom left: (e)* 10 iterations. *Bottom right: (f)* 30 iterations.

3.1.9.3 Smoothing Fluid Dynamics Data

Our second example in Figure 3.8 uses a deformation tensor set from computational fluid dynamics. Here, the eigenvalues of the tensors are of different signs. The structure-preserving smoothing effect of the median filter is again visible.

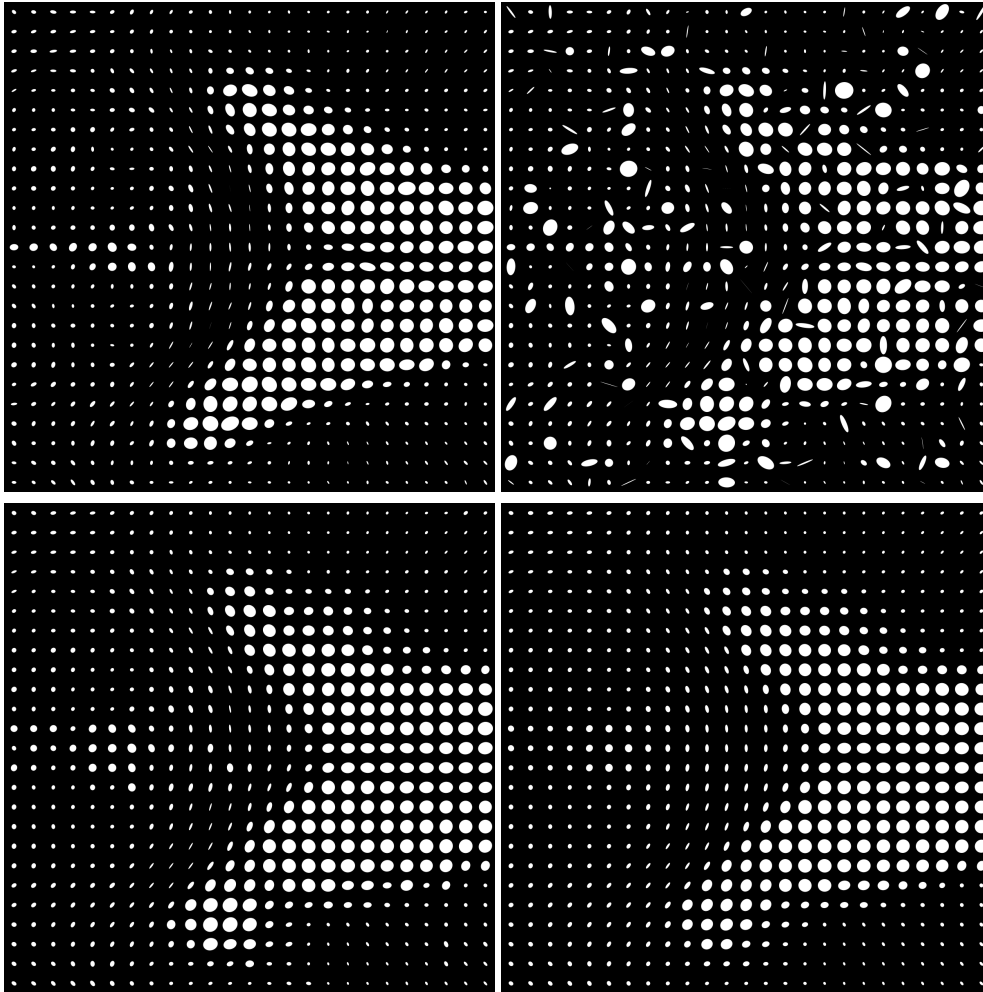


Figure 3.5: Frobenius median filtering of 2D DTMRI data visualised by ellipses. *Top left: (a)* Detail of the original image, Figure 3.4 (a), featuring corpus callosum (centre) and part of the ventricle. *Top right: (b)* Corresponding detail from the noisy image, Figure 3.4 (b). *Bottom left: (c)* Detail from Frobenius median filtered image, 1 iteration, 3×3 stencil. *Bottom right: (d)* 3 iterations, 3×3 stencil.

Remember that we have to use here the deformation-based ellipse visualisation in which a zero matrix is represented as a circle of medium size (compare again Section 1.4.3).

3.1.9.4 Robust Structure Estimation

As a discontinuity-preserving matrix smoother, the matrix median can be used to smooth orientation information that is extracted from textured images via structure tensors. This application has been exposed in [20, 217].

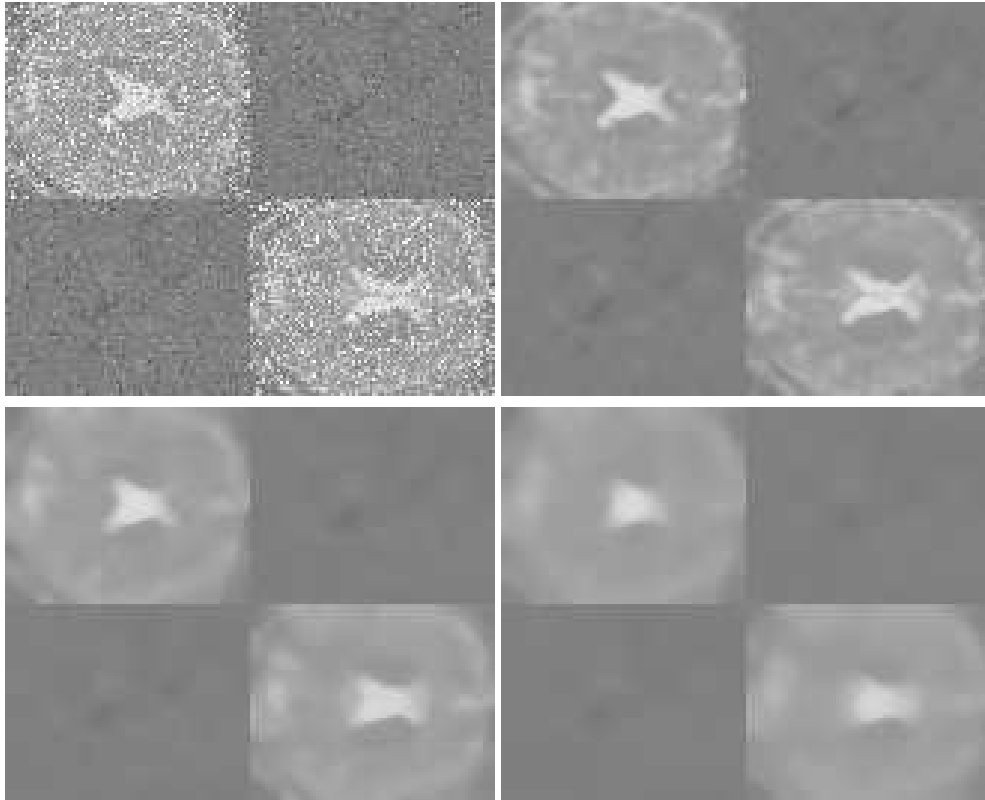


Figure 3.6: Frobenius median filtering of 2-D DTMRI data. *Top left: (a)* Noisy image from Figure 3.4 (b). *Top right: (b)* Filtered by Frobenius median, 3×3 stencil, 3 iterations. *Bottom left: (c)* 5×5 stencil, 3 iterations. *Bottom right: (d)* 7×7 stencil, 3 iterations.

Structure tensors [91] are computed by Gaussian smoothing of the outer product matrices $\nabla u \nabla u^T$ of an image u . They encode local orientation estimation integrated within a neighbourhood on the scale of the Gaussian which is used.

A matrix median filtering step can now be employed for a robust filtering of these structure tensors. This is demonstrated by Figure 3.9. Pursuing this idea further, the Gaussian smoothing can even be omitted; one then applies the median filtering directly to the rank one outer product matrices. When processing structure information from images with no or moderate noise, see e.g. Figure 3.10 a favourable smoothing is achieved which keeps discontinuities in the orientation field fairly sharp.

For images contaminated with stronger noise, see Figure 3.11, the quality of the results is still less satisfactory. Improvements of the orientation estimation are made using two modifications which can be used separately or combined. First, since we are only interested in directional information, the gradients (or, equivalently, the outer product matrices) can be normalised before median filtering. Second, median filtering itself can be iterated. The experiments in Figure 3.12 reveal that in case of

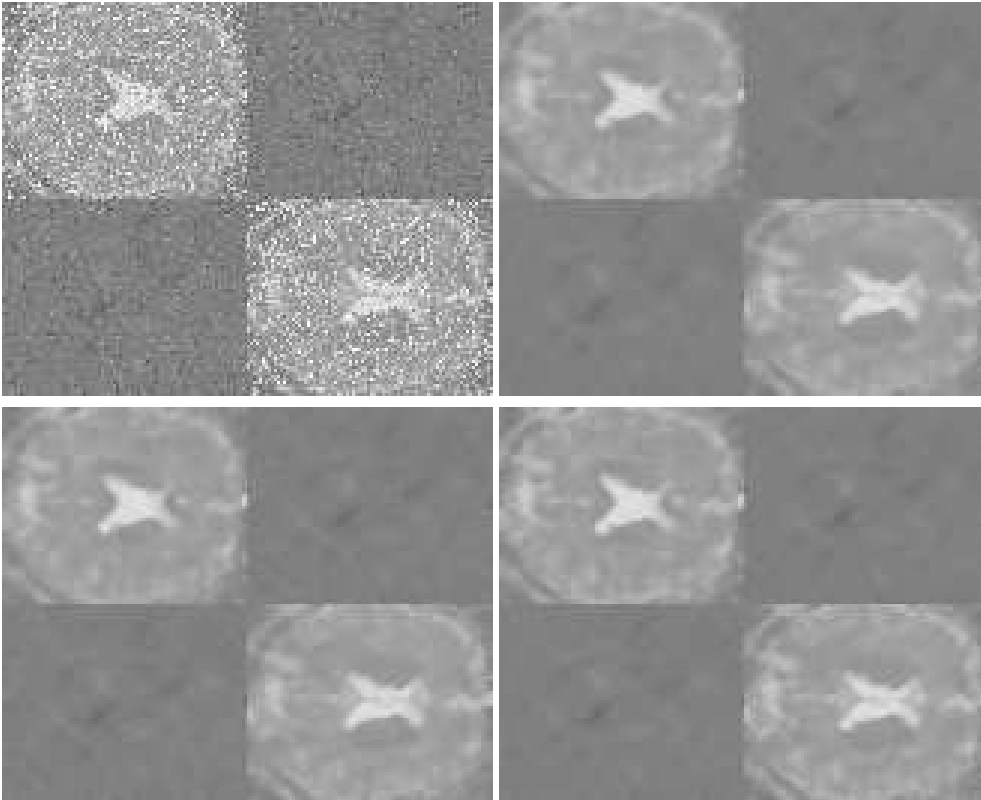


Figure 3.7: Median filtering of 2-D DTMRI data with different norms. *Top left: (a)* Noisy image from Figure 3.4 (b). *Top right: (b)* Filtered by Frobenius median, 3×3 stencil, 3 iterations. *Bottom left: (c)* Nuclear median, 3×3 stencil, 3 iterations. *Bottom right: (d)* Spectral median, 3×3 stencil, 3 iterations.

Table 3.1: Average angular errors (AAE) measured in orientation estimation. Method-specific parameters in brackets include stencil diameter and iteration count for median, m and s (see [199]) for Boomgaard–Weijer tensor. From [225].

Method	AAE undisturbed	AAE impulse noise	AAE Gaussian noise
gradient direction	3.387°	20.612°	31.429°
Frobenius median	1.591° (7, 1)	1.914° (9, 4)	3.207° (9, 5)
Frobenius median, norm.	1.312° (7, 1)	1.655° (5, 5)	3.434° (15, 4)
Boomgaard–Weijer	1.634° (0.1, 3)	1.489° (0.05, 5)	3.657° (0.05, 9)

impulse noise, each of these ideas is capable of sharpening the discontinuity. For Gaussian noise, iterated median filtering gives the greater gain in performance. The combination in this case does not pay off significantly. Table 3.1 juxtaposes quality measurements based on average angular errors for the different methods.

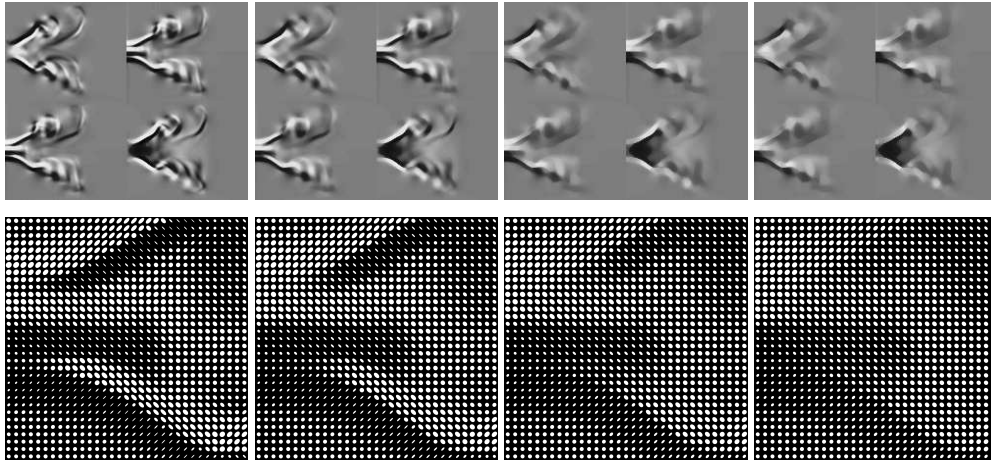


Figure 3.8: Frobenius median filtering of a tensor field containing indefinite matrices. The data are deformation tensors originating from a fluid dynamics simulation. *Left to right: (a)* Initial data, 124×101 pixels. *(b)* 10 iterations, 3×3 stencil. *(c)* 100 iterations. *(d)* 1000 iterations. *Bottom, left to right: (e)–(h)* Detail from the top row visualised by ellipses. Adapted from [219].

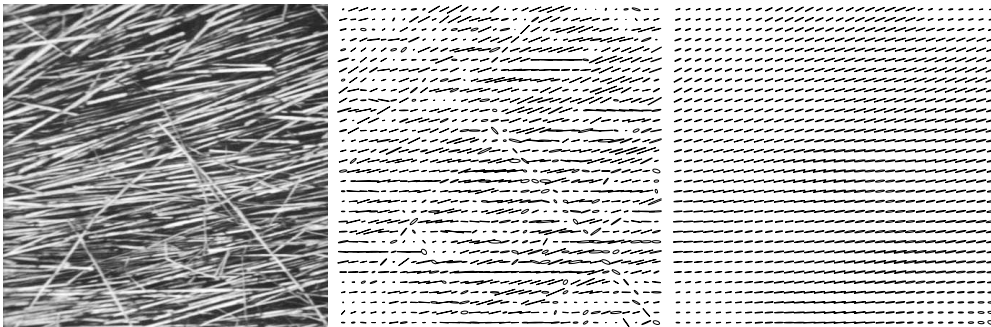


Figure 3.9: *Left to right: (a)* Image containing oriented texture with inhomogeneities. *(b)* Structure tensors computed by smoothing the outer products $\nabla u \nabla u^T$ with 15×15 Gaussian. Gradients have been calculated by 3×3 derivative-of-Gaussian filtering. The final matrix field has been subsampled for visualisation. *(c)* Result of Frobenius median filtering of (b) with 7×7 stencil, subsampled. Image: F. Becker, from [217].

Robust structure tensors. To end this section, we want to point out another aspect. The classical structure tensor smoothes outer product matrices by means of the Gaussian scale space which is simple and efficient but insensitive to features. In [211], compare also [42], Weickert and Brox have replaced Gaussian smoothing, which is in fact a linear diffusion process, by a feature-preserving nonlinear diffusion process, yielding a *nonlinear* structure tensor. Assigning the role of the smoothing process to iterated median filtering, which also constitutes a scale space, stands in analogy to this procedure and can be seen as construction of a *robust* structure tensor.



Figure 3.10: *Left to right:* (a) Synthetic image with oriented textures, inspired by [199]. (b) Local orientations computed via derivatives of Gaussians. Orientations have been mapped to grey-values. Note that the orientations represented by black and white are close neighbours. (c) Orientations after median filtering of the orientation matrices with Frobenius norm and a disk-shaped structure element of diameter 7. (d) Same with structure element of diameter 9. (e) Spectral norm median filtering, diameter 9. Image: F. Becker, from [217].

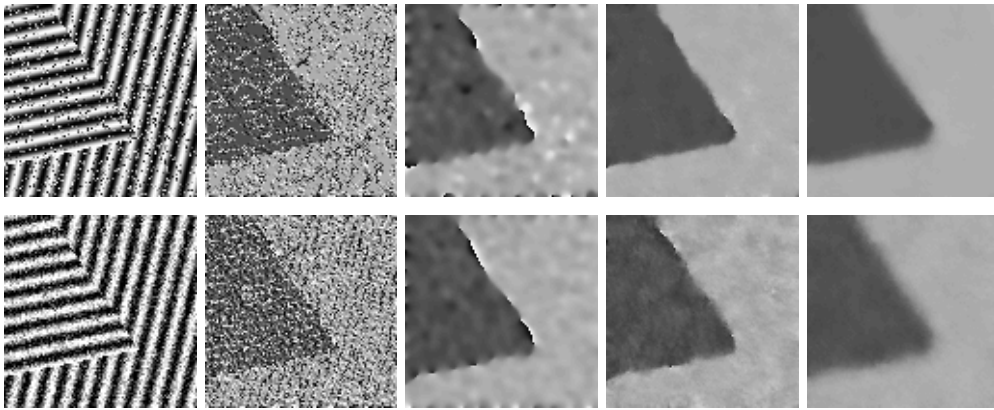


Figure 3.11: *Top, left to right:* (a) Test image with 20% impulse noise. (b) Orientation field of (a). (c) Structure tensor orientation obtained by Gaussian smoothing of the outer product matrices with standard deviation 19. (d) Same after median filtering with Frobenius norm and disk-shaped structure element of diameter 9. (e) Median filtering of (a) with Frobenius norm and disk-shaped structure element of diameter 19. *Bottom, left to right:* (f) Test image perturbed by Gaussian noise of standard deviation 0.2 (where grey-values vary between 0 and 1). (g) Orientation field of (f). (h) Structure tensor orientation as in (c). (i) Median filtering as in (d). (k) Median filtering as in (e). Image: F. Becker, from [217].

The notion of robust structure tensor has also been used by Boomgaard and Weijer in [199], see also [42]. They propose the minimisation of an objective function which leads to a (noniterated) weighted median, compare Sec. 3.2.1 below. Since the weights are defined by a Gaussian, the Boomgaard–Weijer tensor in fact combines median and diffusion operations in one filter. We include orientation estimates with the Boomgaard–Weijer tensor in Figure 3.12 and Table 3.1.

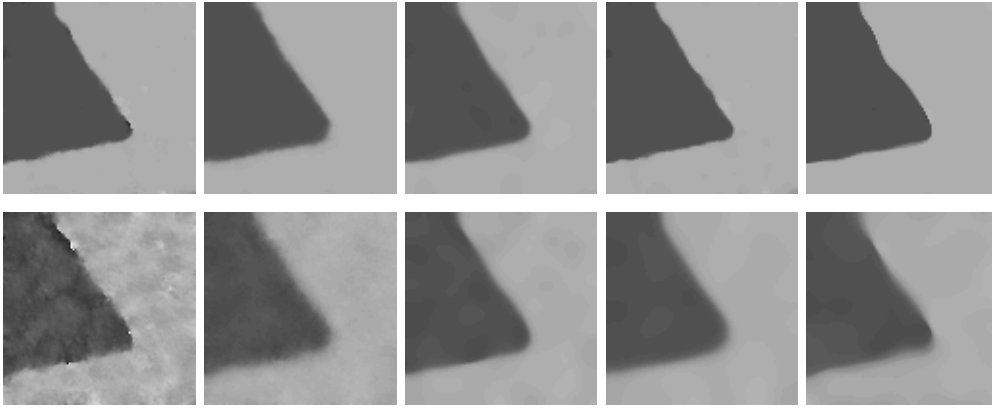


Figure 3.12: *Top row*: Modified local orientation filtering for the impulse-noise image, Figure 3.11 (a). *Left to right*: (a) Frobenius median filtering of normalised outer product matrices with disk-shaped stencil of diameter 9. (b) As (a) but with stencil of diameter 19. (c) Four iterations of the median filter from Figure 3.11 (d). (d) Five iterations of median filter with normalisation, stencil diameter 5. (e) Orientation estimate from the Boomgaard-Weijer robust structure tensor, parameters (see [199]) $m = 0.05, s = 5$. *Bottom, left to right*: Filtering of Figure 3.11 (f). (f) Frobenius median filtering with normalisation and stencil of diameter 9. (g) Same with diameter 19. (h) Median filtering as in Figure 3.11 (i), five iterations. (i) Four iterations of median filter with normalisation, stencil diameter 15. (k) Boomgaard and Weijer's robust structure tensor, $m = 0.05, s = 9$. Image: F. Becker, from [217].

3.2 Median-Related Filters

In this section, we discuss two close relatives of the median concept that can easily be obtained by slight variations of the same minimisation idea. Due to the close analogy, the convexity and uniqueness details need not be discussed again since they transfer in all cases straightforward from the corresponding median definitions (with the same matrix norm).

3.2.1 Weighted Median

As demonstrated before, matrix median filtering allows an efficient and edge-preserving denoising. However, fine details which are smaller than the stencil size still experience a degradation even by a single iteration of median filtering. When denoising images which contain only moderate amounts of noise, the preservation of small details can be improved.

We achieve this by using weighted medians. Unweighted scalar median filtering changes each pixel which has not exactly the middle value within its neighbourhood. If in-

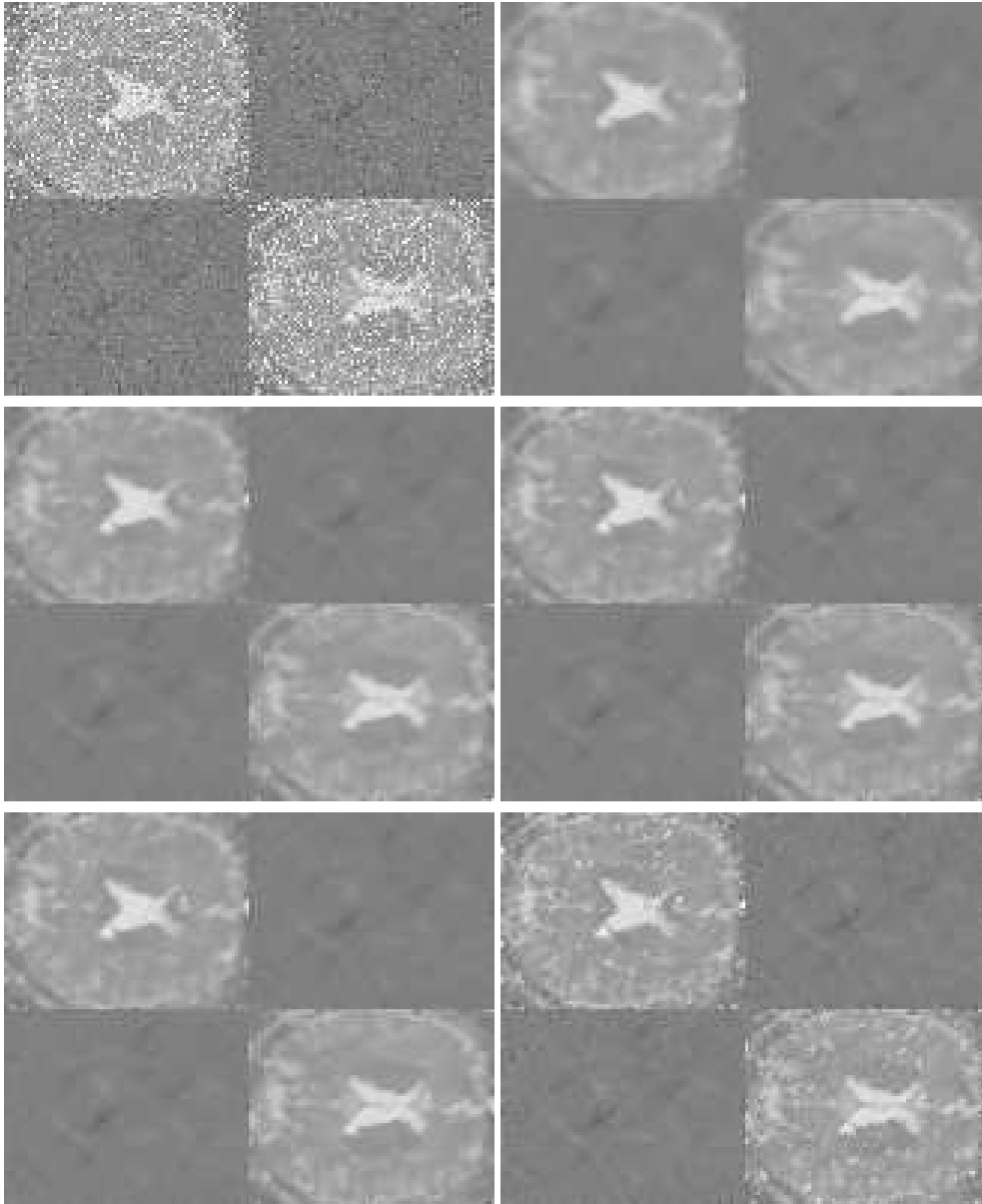


Figure 3.13: Weighted Frobenius median. The noisy DTMRI image has been filtered using 3 iterations of a 3×3 stencil with different weights w of the central pixel. The remaining pixel weights were fixed to 1. *Top left: (a)* The noisy image from Figure 3.4 (a). *Top right: (b)* Filtered by Frobenius median, i.e. $w = 1$. *Middle left: (c)* Weighted with central pixel weight $w = 2$. *Middle right: (d)* $w = 2.2$. *Bottom left: (e)* $w = 2.5$. *Bottom right: (f)* $w = 5$.

stead the central pixel is repeated more than once within the ordered sequence, its value survives even if it is just close to the middle position. Only pixels whose values are close to the extrema within their neighbourhood are treated as outliers and

therefore changed. The transfer of this idea to the energy minimisation formulation is obvious and is formulated in the following definition right away for matrices.

Definition 3.2 *Given a tuple $S = (A_1, \dots, A_n)$ of $d \times d$ matrices, a vector of nonnegative weights $w = (w_1, \dots, w_n)$ and a norm $\|\cdot\|$. The weighted median $\text{med}(S, w)$ is defined as the minimiser of*

$$E_{(S,w)}(X) = \sum_{i=1}^n w_i \|X - A_i\|. \quad (3.28)$$

Figure 3.13 demonstrates denoising of tensor images by weighted matrix median filtering. We use a 3×3 stencil in which the weight of the central pixel is varied. It can be seen that fine structures can be retained that are removed by unweighted median filtering even with small stencils. The admissible weight for the central pixel depends sensitively on the noise level. In our noisy test image, a weight of 2 or slightly above for the central pixel considerably enhances structure preservation while higher weights lead directly to stronger noise.

3.2.2 Matrix-Valued Quantile Filtering

The possibility to transfer the notion of median, thus a 50%-quantile, to matrix data motivates us to check whether even other quantiles can be defined for this type of data. Indeed, the α -quantile $\text{qu}_\alpha(S)$, $0 < \alpha < 1$, of a real data tuple $S = (a_1, \dots, a_n)$ admits a characterisation by a minimisation property similar to that for the median.

3.2.2.1 Scalar-Valued Quantiles

One has that $\text{qu}_\alpha(S)$ minimises the convex function

$$E_{S,\alpha}(x) := \sum_{i=1}^n f_\alpha(x - a_i) \quad (3.29)$$

where $f_\alpha(z)$ is a piecewise linear but asymmetric (except for $\alpha = 1/2$) function replacing the absolute value,

$$f_\alpha(z) := |z| + (1 - 2\alpha)z = \begin{cases} (2 - 2\alpha)|z|, & z \geq 0, \\ 2\alpha|z|, & z < 0. \end{cases} \quad (3.30)$$

3.2.2.2 Matrix-Valued Quantile Definition

In defining matrix-valued quantiles, we require again the properties of scaling and rotational invariance as well as the embedding property for the scalar-valued quantiles.

The way of generalising is mostly analogous to the median case. However, a matrix equivalent of f_α has to be used.

Definition 3.3 Let $S = (A_1, \dots, A_n)$ be a tuple of symmetric $d \times d$ matrices, $0 < \alpha < 1$, and let a norm $\|\cdot\|$ be given. The α -quantile $\text{qu}_\alpha(S)$ of S w.r.t. $\|\cdot\|$ is defined as minimiser of the convex function

$$E_{S,\alpha}(X) := \sum_{i=1}^n \|f_\alpha(X - A_i)\| \quad (3.31)$$

with the function f_α defined in (3.30).

As usual, the operation of f_α on a symmetric matrix Y is defined by action on the eigenvalues. More explicitly, if $Y = Q \text{diag}(\lambda_1, \dots, \lambda_d) Q^T$ with orthogonal Q , then

$$f_\alpha(Y) := Q \text{diag}(f_\alpha(\lambda_1), \dots, f_\alpha(\lambda_n)) Q^T. \quad (3.32)$$

For $\|\cdot\|$ one might consider again nuclear, Frobenius and spectral norm. However, the necessity to apply f_α to the matrices by diagonalisation prevents any generalisation of this quantile definition to other than symmetric square matrices.

3.2.2.3 Quantile Limits for $\alpha \rightarrow 0$ and $\alpha \rightarrow 1$

Scalar α -quantiles can be considered to include the minimum and maximum of a tuple of real numbers as special cases for $\alpha = 0$ and $\alpha = 1$. It needs, however, to be noticed that the minimisation characterisation via minimisation of (3.29) is insufficient for $\alpha = 0$ or $\alpha = 1$ since e.g. $E_{S,0}(x)$ is equally minimised by all lower bounds of the given scalar data. Similarly, the characterisation of matrix-valued quantiles from Definition 3.3 becomes deficient for $\alpha = 0$ or $\alpha = 1$, admitting as minimisers all X for which $X - A_i$ are uniformly negative (positive) semidefinite. A proper definition therefore requires an additional criterion that selects among all these lower bounds.

Recap of matrix supremum and infimum definitions. It is worth comparing this with the situation in [47, 48] where supremum and infimum notions for matrix sets were established in order to introduce morphological filters. We recall first the orderings used there to characterise upper bounds of a given set of matrices:

- In [48], the so-called Loewner (semi-)ordering \preceq_1 was used which is defined for symmetric matrices A, B as follows:

$$A \preceq_1 B \quad \text{if and only if } B - A \text{ is positive semidefinite.} \quad (3.33)$$

- In [47, Sec. 4], a semi-ordering \preceq_2 for positive semidefinite symmetric matrices A, B was introduced by

$$A \preceq_2 B \quad \text{if and only if the ellipse } x^T A^{-2} x = 1 \quad (3.34)$$

$$\text{is contained in the ellipse } x^T B^{-2} x = 1.$$

At first glance, these definitions look fairly different. Notice, however, that \preceq_2 was introduced by reference to the ellipses

$$x^T A^{-2} x = 1 \quad (3.35)$$

(cf. (1.5)) which underly the standard visualisation in Section 1.4.3. As we remarked there, the ellipse

$$x^T A^{-1} x = 1 \quad (3.36)$$

(cf. (1.6)) stands in a more natural relation to the symmetric positive definite matrix A as it represents in the case of diffusion tensors an isosurface for the probability distribution of diffused particles. In order to obtain a physically more meaningful semi-ordering relation, it is therefore natural to replace the ellipses (3.35) by (3.36) in (3.34), leading to the definition

$$A \preceq_{1'} B \quad \text{if and only if the ellipse } x^T A^{-1} x = 1 \quad (3.37)$$

$$\text{is contained in the ellipse } x^T B^{-1} x = 1$$

for symmetric positive semidefinite matrices A, B .

The definition can easily be restated for symmetric matrices without positive definiteness condition by using the deformation representation from Section 1.4.3, in which deformed unit circles take the role of the ellipses to be related:

$$A \preceq_{1''} B \quad \text{if and only if the ellipse } x^T (I + \varepsilon A)^{-2} x = 1 \quad (3.38)$$

$$\text{is contained in the ellipse } x^T (I + \varepsilon B)^{-2} x = 1$$

$$\text{for all } \varepsilon \in (0, \varepsilon^*) \text{ with a small } \varepsilon^* > 0.$$

We will now prove the following equivalence that clarifies the geometric motivation of \preceq_1 and its close relation to the earlier definition \preceq_2 .

Proposition 3.7 *Let A, B be symmetric $d \times d$ matrices.*

- The relation $A \preceq_1 B$ holds if and only if $A \preceq_{1''} B$ holds.*
- If A, B are positive definite, then $A \preceq_1 B$ holds if and only if $A \preceq_{1'} B$ holds.*

In the proof, we will use the following statement.

Lemma 3.8 *Let A, B be symmetric positive definite $d \times d$ matrices. Then $B - A$ is positive semidefinite if and only if $A^{-1} - B^{-1}$ is positive semidefinite.*

Proof. Assume the converse were true, i.e., without loss of generality, $A^{-1} - B^{-1}$ were positive semidefinite but $B - A$ were not.

Then we would have that

$$x^T(A^{-1} - B^{-1})x \geq 0 \quad \text{for all } x \quad (3.39)$$

and

$$y^T(A - B)y > 0 \quad \text{for some } x. \quad (3.40)$$

In this case, let $z := A^{1/2}y$ such that (3.40) becomes

$$z^T(I - A^{-1/2}BA^{-1/2})z > 0 \quad (3.41)$$

while applying (3.39) to $x := A^{1/2}z$ yields

$$z^T(I - A^{1/2}B^{-1}A^{1/2})z \geq 0. \quad (3.42)$$

Combining (3.41) and (3.42) we find

$$\begin{aligned} 0 &> z^T(A^{-1/2}BA^{-1/2} - 2I + A^{1/2}B^{-1}A^{1/2})z \\ &= z^T(B^{1/2}A^{-1/2} - B^{-1/2}A^{1/2})^T(B^{1/2}A^{-1/2} - B^{-1/2}A^{1/2})z \end{aligned} \quad (3.43)$$

contradicting the positive semidefiniteness of the inner product. \square

Proof of the Proposition. Consider first the case (b). In fact one has for positive definite A, B that $A \preccurlyeq_1 B$, i.e., the ellipse $x^T A^{-1} x = 1$ is completely contained in $x^T B^{-1} x = 1$ if for every x with $x^T A^{-1} x = 1$ the inequality $x^T B^{-1} x \leq 1$ holds, and by rescaling generally

$$x^T B^{-1} x \leq x^T A^{-1} x \quad \text{for all } x, \quad (3.44)$$

which is equivalent to

$$x^T(B^{-1} - A^{-1})x \leq 0 \quad \text{for all } x, \quad (3.45)$$

i.e., positive semidefiniteness of $A^{-1} - B^{-1}$. Lemma 3.8 ensures equivalence to $A \preccurlyeq_1 B$.

For (a), notice that for sufficiently small $\varepsilon > 0$, both $I + \varepsilon A$ and $I + \varepsilon B$ are positive definite. Similarly as in the previous proof, the geometric inclusion condition that defines $\preccurlyeq_{1''}$ is equivalent to

$$x^T((I + \varepsilon A)^{-2} - (I + \varepsilon B)^{-2})x \geq 0 \quad (3.46)$$

and by Lemma 3.8 to

$$\begin{aligned} 0 &\leq x^T((I + \varepsilon B)^2 - (I + \varepsilon A)^2)x \\ &= x^T \underbrace{(2\varepsilon(B - A) + \varepsilon^2(B^2 - A^2))}_{=: C} x. \end{aligned} \quad (3.47)$$

As $\varepsilon > 0$ goes to 0, the sign of $x^T C x$ is determined by $x^T(B-A)x$, except if $x^T(B-A)x = 0$.

If $B \neq A$, the latter holds only in a subspace (possibly 0), which contains no open set of \mathbb{R}^d ; continuity then ensures that the sign of $x^T C x$ for these x cannot be opposite to a uniform sign for all other x , such that for sufficiently small ε , the matrix C is positive semidefinite if and only if $B-A$ is. On the other hand, if $B = A$, then $B^2 = A^2$, too, such that C vanishes as well. In all cases, we have that $A \preceq_{1''} B$ holds if and only if $A \preceq_1 B$ holds. \square

Let us for brevity denote by $\text{Upp}_1(S)$ and $\text{Upp}_2(S)$ the sets of upper bounds of the given matrix tuple S with respect to the partial orderings \preceq_1 and \preceq_2 , respectively, i.e.,

$$\text{Upp}_p(S) := \{X \in \mathcal{S}^d \mid A_i \preceq_p X \text{ for all } i\}, \quad p = 1, 2. \quad (3.48)$$

In both cases [47, 48] an additional criterion had to be used to select among all upper bounds from $\text{Upp}_2(S)$, $\text{Upp}_1(S)$, respectively. In [47] a lexicographic ordering of eigenvalues plays this role, such that a supremum $\text{sup}_2(S)$ for a set S of positive definite symmetric $d \times d$ matrices is defined as follows:

$$\text{sup}_2(S) := \underset{X \in \text{Upp}_2(S)}{\text{argmin}} (\lambda_1(X), \dots, \lambda_d(X)), \quad (3.49)$$

where the eigenvalue tuples are written in decreasing order $\lambda_1 \geq \dots \geq \lambda_d > 0$, and argmin is taken w.r.t. their lexicographic ordering.

Looking at the leading eigenvalue λ_1 , it becomes clear that the lexicographic ordering means to select a matrix with minimal spectral norm, with sensible handling of non-unique cases. Unfortunately, this definition cannot avoid discontinuities in some situations.

In [48], the unique matrix with smallest trace is selected from the upper bound set:²

$$\text{sup}_1(S) := \underset{X \in \text{Upp}_1(S)}{\text{argmin}} \text{tr}(X) \quad (3.50)$$

In the case of positive semidefinite matrices, this can also be interpreted as minimising the nuclear norm.

Matrix suprema and infima as limits of quantiles. In the quantile framework, the limit process $\alpha \rightarrow 1-0$ ($\alpha \rightarrow 0+0$) lends itself as a way to disambiguate the supremum (infimum). Depending which norm is chosen for the α -quantiles, different suprema arise as limits.

²In the penumbra cone model of [48], this criterion is encoded in the step of picking among all cones that cover the given set the one whose tip has smallest vertical coordinate in the chosen reparametrisation. This vertical coordinate is exactly the trace of the matrix.

Proposition 3.9 *Let a tuple $S = (A_1, \dots, A_n)$ of symmetric $d \times d$ matrices be given, and denote by $\text{qu}_{1,\alpha}(S)$, $\text{qu}_{2,\alpha}(S)$, $\text{qu}_{\infty,\alpha}(S)$ the α -quantiles of S with respect to the nuclear, Frobenius, and spectral norms, respectively. Then the following statements describe the limits of these quantiles for $\alpha \rightarrow 1-0$:*

(a) *With the nuclear norm, we have*

$$\text{qu}_{1,1}(S) := \lim_{\alpha \rightarrow 1-0} \text{qu}_{1,\alpha}(S) = \underset{X \in \text{Upp}_1(S)}{\text{argmin}} \text{tr}(X), \quad (3.51)$$

i.e., the matrix supremum $\text{sup}_1(S)$ from (3.50). Moreover, there is an $\hat{\varepsilon} = \hat{\varepsilon}_1(n) > 0$ such that $\text{qu}_{1,\alpha}(S) = \text{sup}_1(S)$ for all $\alpha > 1 - \hat{\varepsilon}$.

(b) *Revised 2016-02: This statement (b) in its original form contained an error. The following is a corrected version published first in [Theorem 1][237].*

With the Frobenius norm, we have

$$\text{qu}_{2,1}(S) := \lim_{\alpha \rightarrow 1-0} \text{qu}_{2,\alpha}(S) = \underset{X \in \text{Upp}_1(S)}{\text{argmin}} \sum_{i=1}^n \|X - A_i\|_{(2)}. \quad (3.52)$$

(c) *With the spectral norm, we have*

$$\text{qu}_{\infty,1}(S) := \lim_{\alpha \rightarrow 1-0} \text{qu}_{\infty,\alpha}(S) = \underset{X \in \text{Upp}_1(S)}{\text{argmin}} \sum_{i=1}^n |\lambda_1(X - A_i)| \quad (3.53)$$

where $\lambda_1(Y)$ denotes the largest eigenvalue of the symmetric matrix Y . There exists an $\hat{\varepsilon} = \hat{\varepsilon}_{\infty}(n) > 0$ such that $\text{qu}_{\infty,\alpha}(S) = \text{qu}_{\infty,1}(S)$ for all $\alpha > 1 - \hat{\varepsilon}$.

Remark. It becomes evident that in the nuclear quantile case a robustness property similar to Lemma 3.5 holds, which ensures the independency of the supremum on those $A_i \in S$ for which $X - A_i$ is positive definite. The limit cases of Frobenius and spectral quantiles, in contrast, depend in general on all matrices of S .

We remark also that $\text{Upp}_1(S)$ consists of positive (semi-)definite matrices if S contains at least one positive (semi-)definite matrix, such that preservation of positive (semi-)definiteness is guaranteed in all cases.

Proof. Throughout this proof, we set for abbreviation

$$\lambda_{ji} := \lambda_j(X - A_i), \quad (3.54)$$

where $\lambda_1(Y) \geq \lambda_2(Y) \geq \dots \geq \lambda_d(Y)$ are again the eigenvalues of a symmetric matrix Y in decreasing order.

To prove (a), we start by defining the index sets

$$\begin{aligned} T &:= \{1, \dots, n\} \times \{1, \dots, d\}, \\ T_+(X) &:= \{(i, j) \in T \mid \lambda_{ji} > 0\}, \\ T_-(X) &:= \{(i, j) \in T \mid \lambda_{ji} < 0\}. \end{aligned} \quad (3.55)$$

The energy function (3.31) with nuclear norm for $\alpha := 1 - \varepsilon$ becomes

$$\begin{aligned} E_{1-\varepsilon}(X) &= \sum_{(i,j) \in T_-(X)} 2(1-\varepsilon)|\lambda_{ji}| + \sum_{(i,j) \in T_+(X)} 2\varepsilon\lambda_{ji} \\ &= 2\varepsilon \sum_{(i,j) \in T} \lambda_{ji} + 2 \sum_{(i,j) \in T_-(X)} |\lambda_{ji}| \\ &= 2\varepsilon \sum_{i=1}^n \text{tr}(X - A_i) + 2 \sum_{(i,j) \in T_-(X)} |\lambda_{ji}| \\ &= 2\varepsilon n \text{tr}(X) + 2 \sum_{(i,j) \in T_-(X)} |\lambda_{ji}| - 2\varepsilon n \text{tr}(\bar{A}). \end{aligned} \quad (3.56)$$

Let $X^* = X^*(\varepsilon)$ be the minimiser of $E_{1-\varepsilon}$, and δ be any real number with

$$0 < \delta < \min_{(i,j) \in T_-(X^*)} |\lambda_{ji}|. \quad (3.57)$$

Then we have

$$\begin{aligned} E_{1-\varepsilon}(X^* + \delta I) &= 2\varepsilon n \text{tr}(X^* + \delta I) + 2 \sum_{(i,j) \in T_-(X^*)} |\lambda_{ji} + \delta| - 2\varepsilon n \text{tr}(\bar{A}) \\ &= E_{1-\varepsilon}(X^*) + 4\varepsilon \delta n - 2\delta |T_-(X^*)| \end{aligned} \quad (3.58)$$

with $|T_-(X^*)|$ denoting the cardinality of $T_-(X^*)$. Since by the minimiser property of X^* the value $E_{1-\varepsilon}(X^* + \delta I)$ must be greater or equal $E_{1-\varepsilon}(X^*)$, we conclude that $4\varepsilon \delta n - 2\delta |T_-(X^*)| \geq 0$, hence

$$|T_-(X^*)| \leq 2\varepsilon n. \quad (3.59)$$

If therefore $\varepsilon < 1/2n$, it follows that $T_-(X^*)$ is empty, thus $X^* \in \text{Upp}_1(S)$. Since for $X \in \text{Upp}_1(S)$, equation (3.56) simplifies to

$$E_{1-\varepsilon}(X) = 2\varepsilon \text{tr}(X) - 2\varepsilon n \text{tr}(\bar{A}), \quad (3.60)$$

we see that

$$X^*(\varepsilon) = \underset{X \in \text{Upp}_1(S)}{\text{argmin}} \text{tr}(X) \quad \text{for } \varepsilon < \frac{1}{2n} \quad (3.61)$$

which proves statement (a) with $\hat{\varepsilon}_1(n) = 1/2n$.

(Revised 2016-02: The proof of part (b) at this place has been deleted as it was flawed. The proof for the corrected statement (b) is found in [237].)

It remains to prove statement (c). Here we choose the index sets

$$\begin{aligned} T &:= \{1, \dots, n\}, \\ T_+(X) &:= \{i \in T \mid \lambda_{di} \geq 0 \text{ or } \varepsilon \lambda_{1i} \geq -(1-\varepsilon)\lambda_{di} \geq 0\}, \\ T_-(X) &:= \{i \in T \mid \lambda_{1i} < 0 \text{ or } (1-\varepsilon)\lambda_{di} < -\varepsilon \lambda_{1i} < 0\}. \end{aligned} \quad (3.69)$$

Thus, $T_+(X)$ indexes those matrices A_i for which the largest positive eigenvalue determines the spectral norm of $f_{1-\varepsilon}(X - A_i)$, while $T_-(X)$ indexes those for which the smallest negative eigenvalue plays this role.

With the spectral norm, (3.31) then takes the form

$$E_{1-\varepsilon}(X) = 2(1-\varepsilon) \sum_{i \in T_-(X)} |\lambda_{di}| + 2\varepsilon \sum_{i \in T_+(X)} |\lambda_{1i}|. \quad (3.70)$$

Denote again by X^* the minimiser of $E_{1-\varepsilon}$, and choose a small positive δ such that $T_+(X^* + \delta I) = T_+(X^*)$, $T_-(X^* + \delta I) = T_-(X^*)$. (This is achieved if on replacing X^* with $X^* + \delta I$, all negative eigenvalues λ_{ji} stay negative, and for no indefinite $X^* - A_i$, the order of $\varepsilon \lambda_{1i}$ and $-(1-\varepsilon)\lambda_{di}$ changes.) Then we have

$$\begin{aligned} E_{1-\varepsilon}(X^* + \delta I) &= 2(1-\varepsilon) \sum_{i \in T_-(X^*)} (|\lambda_{di}| - \delta) + 2\varepsilon \sum_{i \in T_+(X^*)} (\lambda_{1i} + \delta) \\ &= E_{1-\varepsilon}(X^*) - 2(1-\varepsilon)\delta |T_-(X^*)| + 2\varepsilon \delta |T_+(X^*)|, \end{aligned} \quad (3.71)$$

and this has to be greater or equal $E_{1-\varepsilon}(X^*)$. Hence,

$$|T_-(X^*)| \leq \frac{\varepsilon}{1-\varepsilon} |T_+(X^*)| \leq \frac{\varepsilon}{1-\varepsilon} n. \quad (3.72)$$

If $\varepsilon < \frac{1}{n+1}$, the last expression is less than 1, such that $T_-(X^*)$ must be empty, i.e. $X^* \in \text{Upp}_1(S)$. For $X \in \text{Upp}_1(S)$, equation (3.70) simplifies to

$$E_{1-\varepsilon}(X) = 2\varepsilon \sum_{i=1}^n |\lambda_{1i}|, \quad (3.73)$$

therefore $X^* = \operatorname{argmin}_{X \in \text{Upp}_1(S)} \sum_{i=1}^n |\lambda_{1i}|$. Statement (c) follows, with $\hat{\varepsilon}_\infty(n) = \frac{1}{n+1}$. \square

Matrix infima, which can be defined analogously by the limit process $\alpha \rightarrow +0$, also obey statements corresponding to those of Proposition 3.9. Preservation of positive (semi-)definiteness, however, cannot be guaranteed in all cases.³

³For example, the analog of $\text{sup}_1(S)$ would be $\text{inf}_1(S)$ which is the matrix X of greatest trace such that all $A_i - X$ are positive semidefinite. This matrix can happen to be indefinite. In [48], it is therefore proposed to use instead $(\text{sup}_1(A_1^{-1}, \dots, A_n^{-1}))^{-1}$, i.e., the matrix X for which all $A_i - X$ and X are positive semidefinite and for which $\text{tr}(X^{-1})$ is smallest.

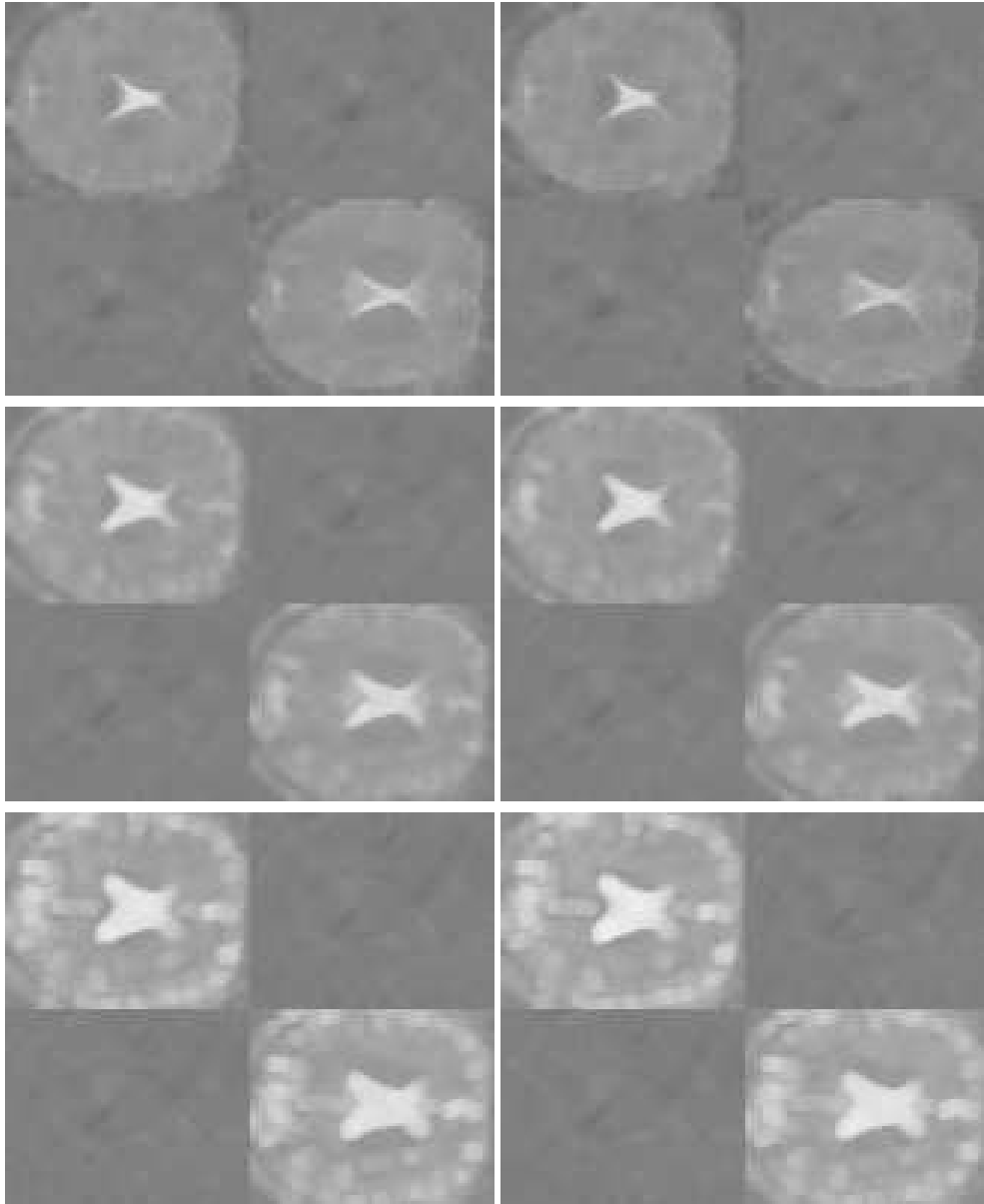


Figure 3.14: Matrix-valued α -quantiles. The DTMRI image from Figure 3.4 (a) has been filtered with a 5×5 stencil, 1 iteration. *Top row:* $\alpha = 0.1$. *Middle row:* $\alpha = 0.5$ (i.e. median). *Bottom row:* $\alpha = 0.9$. – *Left column:* Frobenius quantiles. *Right column:* Nuclear quantiles.

Experiments. We demonstrate α -quantile filtering of DTMRI data in Figure 3.14. It can be seen that for α close to 0 or 1 the nuclear quantile filters become similar to morphological erosion and dilation operators.

3.3 Midrange Filter

The minimisation approach underlying the matrix median definition can easily be extended to a larger class of local image filters, cf. [219].

3.3.1 Scalar-Valued Midrange

Given a set S of real numbers, its mid-range value is simply the arithmetic mean of their maximum and minimum. A mid-range filter is then obtained by taking the mid-range value of the grey-values within a suitable neighbourhood of a pixel. Mid-range filters are rarely used for denoising purposes since they perform reasonably only in fairly special situations (noise distributions with “thin tails”). They can, however, be used in the construction of more relevant filters.

Interestingly, the mid-range value can be seen as minimiser of the convex function

$$E_S(x) = \max_{i=1,\dots,n} |x - a_i|, \quad (3.74)$$

which gives it a formulation very similar to that of the median.

3.3.2 Multi-Channel Midrange Definition

In analogy to our generalisation of the median from scalar to multi-channel data, the mid-range for multi-channel data is defined as the minimiser of a convex function.

Definition 3.4 *Given a tuple $S = (A_1, \dots, A_n)$ of symmetric matrices, its mid-range value $\text{midr}(S)$ is the minimiser of the convex function*

$$E_S(X) = \max_{i=1,\dots,n} \|X - A_i\| \quad (3.75)$$

with a matrix norm $\|\cdot\|$.

Based on similar requirements as for the median, suitable choices include once more nuclear, Frobenius and spectral norm.

3.3.3 Properties of Multi-Channel Midrange Filters

3.3.3.1 Frobenius Midrange

We start again by considering uniqueness. Since for all matrix norms the energy E_S used in the definition of midrange values is convex, there is either a unique minimiser

or a convex set of minimisers with equal values of E_S . The latter case implies particularly that those matrix norms $\|X - A_i\|$ which realise the maximum in E_S for the minimisers are constant along some straight line segment.

In the case of the Frobenius midrange, the convexity properties of the Frobenius norm discussed in 3.1.3 rule out the existence of such a straight line segment, thereby ensuring uniqueness of Frobenius midrange values in all cases.

Moreover, Proposition 3.2 transfers directly to the Frobenius midrange case, which guarantees that the midrange value of positive semidefinite matrices is also positive semidefinite.

3.3.3.2 Nuclear Midrange

Since the exceptions from strict convexity are significant for the nuclear norm, there are also generic cases in which the nuclear midrange value is not uniquely defined. We sketch such a situation: Let X be a minimiser of E_S for some given data set S . Assume that all those $X - A_i$ whose norms attain the maximum in the definition of E_S are positive or negative definite. Then there is a hyperplane segment around X consisting of matrices X' with $\text{tr}(X') = \text{tr}(X)$ for which the corresponding $X' - A_i$ are also positive or negative definite and for which the same A_i determine the maximum in the definition of E_S . All these X' are equally minimisers for E_S as X is.

As a consequence, the “weak” positive semidefiniteness preservation stated in Proposition 3.4 for the nuclear median translates also to the nuclear midrange.

3.3.3.3 Spectral Midrange

Also the spectral midrange value can be non-unique for certain input sets. This can happen when only two input data values realise the maximum of $\|X - A_i\|$ in (3.75) for the midrange value X . Often, however, the maximum is realised by the corners of a simplex in the input data space, i.e., by four input values (in the case of symmetric 2×2 matrices) or seven input values (in the case of symmetric 3×3 matrices), which determine the midrange value uniquely.

3.3.4 Numerics

3.3.4.1 Gradient Descent Based Method

The gradient descent based algorithm from Section 3.1.7 can easily be adapted to calculate mid-range values. However, due to the maximum operation the gradient at

each location is dominated by one single matrix difference norm $\|X - A_i\|$. This induces additional hypersurface-singularities separating zones of dominance of the A_i . The regularisation procedure described in Section 3.1.7 for the spectral and nuclear norm case can be used to handle these singularities; as a consequence, regularisation is necessary even in the Frobenius mid-range computation. For the spectral and nuclear norms, their specific singularities as discussed in Section 3.1.7 have to be respected as well. Moreover, the step-size control mechanism is indispensable in this case.

3.3.4.2 Convex Optimisation

A representation of the mid-range operator by a convex optimisation problem works along the lines described for the spectral median, with the difference that only a scalar auxiliary variable is needed. For details we refer to [20].

3.3.5 Experimental Evaluation

Mid-range filtering of 2D DTMRI data is shown in Figure 3.15. We display Frobenius and spectral midrange filtering results for 3×3 and 5×5 stencils. The nuclear midrange is less attractive since it suffers from the above-mentioned non-uniqueness problem.

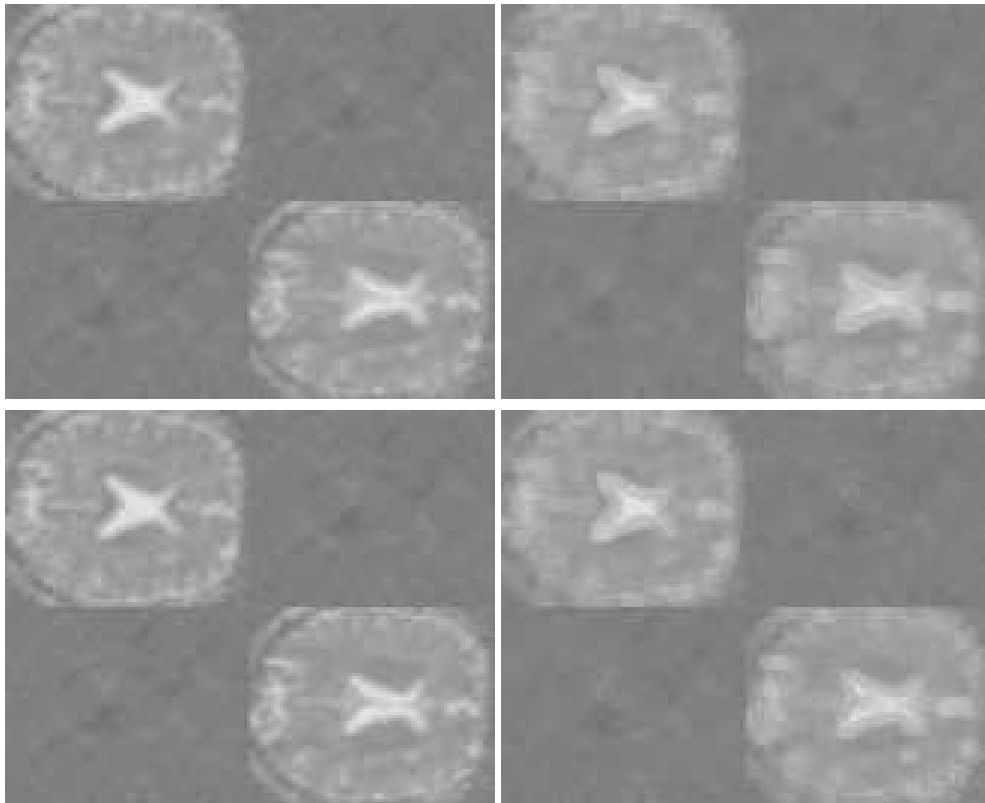


Figure 3.15: Matrix-valued midrange filtering of the DTMRI image from Figure 3.4. *Top left*: Frobenius midrange, 3×3 stencil. *Top right*: Frobenius midrange, 5×5 stencil. *Bottom row*: Same with spectral midrange filter.

3.4 M-Estimators with Arbitrary Power Functions as Penalisers

In this section, we consider a class of filters that generalises and extends median and midrange filters.

3.4.1 Scalar-Valued M-Smothers and M-Estimators

Replacing the distances $|x - a_i|$ in the function (3.1) by some other function of $|x - a_i|$ yields a more general class of nonlinear averages for real numbers. Minimisers of

$$E_S(x) := \sum_{i=1}^n \psi(|x - a_i|) \quad (3.76)$$

are called *M-estimators* [19]. Like the median and mid-range value, M-estimators give rise to local image filters which are denoted as *M-smothers* [194, 232].

An important choice for ψ on which we will concentrate in this work is the p -th power, $\psi(s) = s^p$, $p > 0$.

Special cases of these M-estimators include the median for $p = 1$ but also, in the least-squares case $p = 2$, the arithmetic mean. As limit case for $p \rightarrow \infty$, the mid-range value as well fits into the framework.

The M-estimators for $p < 1$ are more difficult to handle since their objective functions are no longer convex – instead, they have local minima at all input values and are strictly concave in the remainder of the real line.⁴ Nevertheless, the corresponding M-smothers display attractive properties for applications since they exceed the median filter in robustness and are able even to *enhance* edges.

3.4.2 Multi-Channel M-Smothers and M-Estimators

By a straightforward extension of the matrix-valued median and mid-range concepts, we are led to the following definition for matrix-valued M-estimators.

Definition 3.5 *Let $S = (A_1, \dots, A_n)$ be a tuple of symmetric $d \times d$ matrices, and p a positive real number. A symmetric matrix which minimises the convex function*

$$E_S(X) := \sum_{i=1}^n \psi(\|X - A_i\|) \quad (3.77)$$

⁴These M-estimators are often called *modes*. The usual mode value of a set, i.e., the most frequent value, arises in the limit case $p \rightarrow 0$.

with some matrix norm $\|\cdot\|$ and an increasing function $\psi : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is a matrix-valued *M*-estimator for S .

As in the scalar-valued case, we restrict our considerations to the case $\psi(s) = s^p$ with $p > 0$.

Clearly, the matrix-valued median is recovered for $p = 1$. For $p > 1$, there exists a unique minimiser for E_S because of the strict convexity of that function.

As in the scalar case, one faces a more complex situation for $p < 1$ which we discuss here exemplarily for the Frobenius norm.

Lemma 3.10 *Let a set $S = \{A_1, \dots, A_n\}$ of symmetric $d \times d$ matrices be given, and consider the function E_S from (3.77) with Frobenius norm and $\psi(s) = s^p$, $0 < p < 1$. Then each matrix A_i is a local minimum of E_S . Moreover, there is a bound $\varrho > 0$ such that within no ϱ -neighbourhood of any A_i there is another local minimum of E_S .*

Remark. The first part of the statement, i.e. that each of the given matrices A_i is a local minimum of E_S , is in full analogy to the scalar-valued case (see Section 3.4.1). Unlike there, additional minima of the objective function can now exist in regions where E_X is smooth, but they are bounded away from the first type of minima by the second assertion.

Proof. Only the second assertion needs to be proven. Since the gradient magnitude $|\nabla(\|Y - A_j\|^p)|$ grows over all limits when Y approaches the singularity at A_j , there exists for each $p < 1$ and given data set X a radius $\varrho = \varrho(p, X)$ such that, within a ϱ -neighbourhood of A_j , the gradient $\nabla(\|Y - A_j\|^p)$ dominates $\sum_{i \neq j} \nabla(\|Y - A_i\|^p)$ guaranteeing A_j to be the only local minimum of E_X within that neighbourhood. \square

In the design of an *M*-smoother with $p < 1$, the crucial question therefore arises which minimum of E_X should be chosen as the value of the *M*-estimator. We mention two alternatives. The global minimum with its advantage of avoiding any artificial assumptions has the drawback of being highly sensitive to changes in the input data. An alternative is a focussing strategy that starts with the unique minimum at $p = 1$ and tracks its drifting while p decreases. Although this method reduces the set of minima to be considered, instabilities of two kinds are still introduced into it by the way how minima evolve with decreasing p . First, minima can split by bifurcations where no obvious criterion tells which branch to follow. Second, minima can disappear; then the focussing must jump into another minimum, chosen e.g. by gradient descent. Besides that, the focussing method applied in the scalar setting would just trivially lock in at the median.

3.4.3 Numerical Aspects

3.4.3.1 Case $1 < p < \infty$

For M-smoothers with $1 < p < \infty$, the gradient descent algorithm can be applied similarly as for the median, taking care of the necessary regularisations in case of the spectral norm. The nuclear norm does not require regularisations here since the p -th power smoothes out the singularities in the absolute values of eigenvalues.

We mention that for $p > 1$, unlike for $p = 1$, the gradient magnitude is sensitive to re-scaling of the data range and thus contains information about the distance to the minimum. This information could be used to determine the step size instead of the adaptive control mechanism described above. We do not elaborate this further.

3.4.3.2 Case $p < 1$

For $p < 1$ the missing convexity is the dominating problem. Depending on the decision which of the multiple local minima should be the value of the M-estimator (compare end of 3.4.2), different strategies are possible. If one is interested in the global minimum, one option is a grid search within the convex hull of the input data. In Figure 3.16, we present filtering results obtained in this way.

3.5 Multi-Channel Median and Mean Curvature Motion

In 3.1.1 we cited the approximation result established by Guichard and Morel that links scalar-valued median filtering to a PDE-based image filter, namely mean curvature motion (MCM).

A generalisation of mean curvature motion to matrix-valued data has been given in [85, 86], see also [212]. This gives rise to the question whether the median—MCM link can also be generalised to the matrix-valued setting. We conclude the present chapter with an investigation of this question.

3.5.1 Multi-Channel Mean Curvature Motion

Given a scalar-valued image f on a closed domain $\Omega \in \mathbb{R}^2$, mean curvature motion [3, 4, 118] is the image evolution $u : \Omega \times [0, \infty) \rightarrow \mathbb{R}$ described by the equation

$$\partial_t u = |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad \text{on } \Omega \times [0, \infty) \quad (3.78)$$

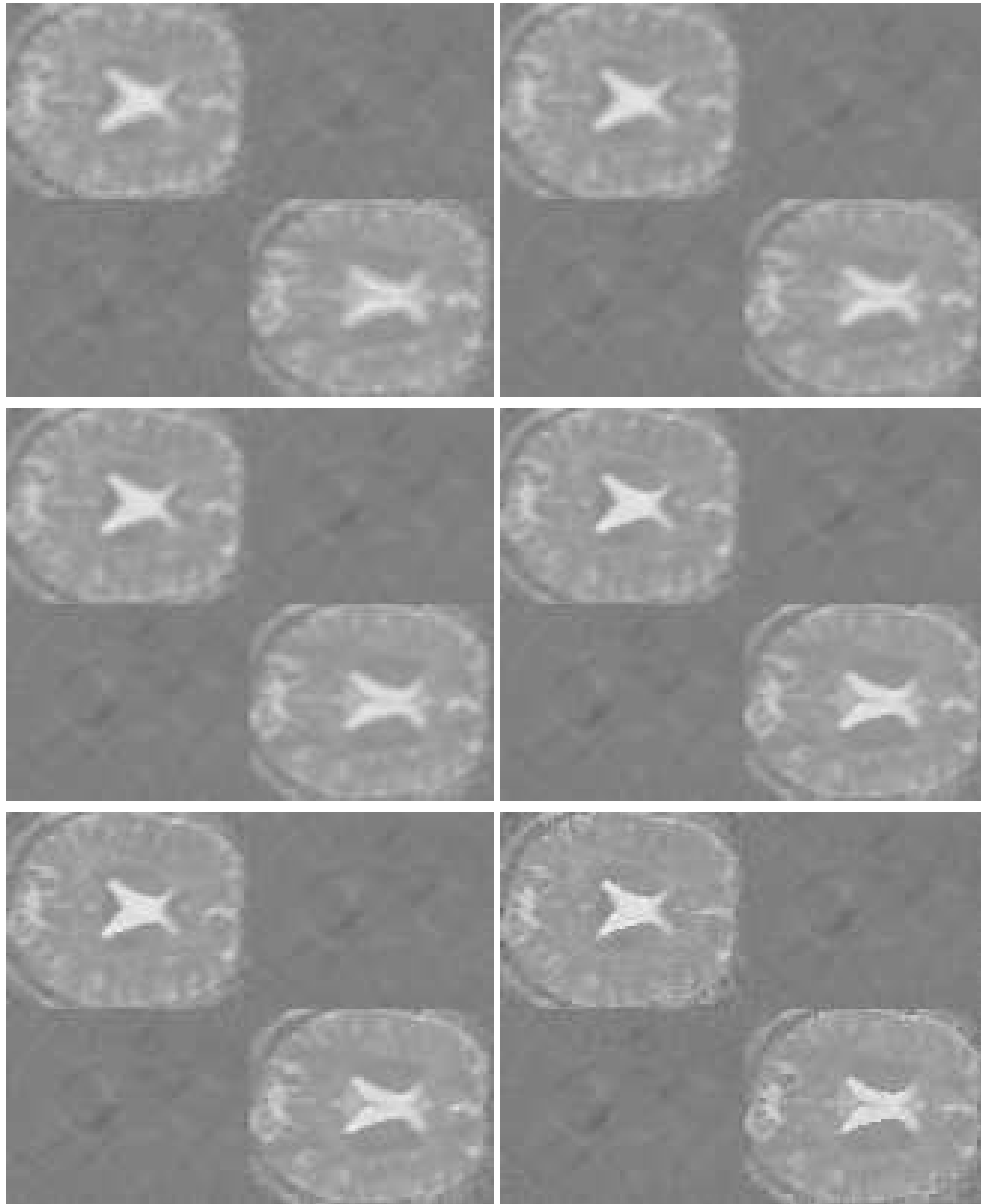


Figure 3.16: Matrix-valued Frobenius M -smoothing of the DTMRI image from Figure 3.4 with penalisers $\psi(s) = s^p$ for different p . One filtering step with 3×3 stencil has been performed in all cases. *Top left:* (a) Frobenius midrange as limiting case $p \rightarrow +\infty$. *Top right:* (b) $p = 3$. *Middle left:* (c) $p = 2$, equivalent to box averaging. *Middle right:* (d) $p = 1$, standard Frobenius median. *Bottom left:* (e) $p = 0.5$, global minimiser in each pixel computed via grid search. *Bottom right:* (f) $p = 0.1$, grid search.

with the initial condition

$$u(x, 0) = f(x) \quad \text{for all } x \in \Omega \quad (3.79)$$

and suitable boundary conditions on $\partial\Omega \times [0, \infty)$.

The process (3.78) can be restated as

$$\partial_t u = \partial_{\xi} \xi u \quad (3.80)$$

where for each (x, t) , the unit vector ξ is chosen such that $v \perp \nabla u(x, t)$.

The latter description underlies the definition of multi-channel mean curvature motion which we will introduce now. This concept goes back to [55] and has been adapted in [85, 86] for matrix-valued images.

Given a multi-channel image $f = (f_k)_{k \in \Gamma}$ with channels enumerated by Γ , we consider the partial differential equation

$$\partial_t u_k = \partial_{v v} u_k \quad (3.81)$$

on a closed domain $\Omega \subset \mathbb{R}^2$ with initial condition

$$u_k(x, 0) = f_k(x) \quad \text{for all } x \in \Omega \quad (3.82)$$

where at each location $x \in \Omega$ and time $t \geq 0$ the unit vector v is an eigenvector of the multi-channel structure tensor (compare 2.3.2)

$$J = J_0(\nabla u_\sigma) := \sum_{k \in \Gamma} \nabla(K_\sigma * u_k) \nabla(K_\sigma * u_k)^T \quad (3.83)$$

corresponding to the smallest eigenvalue of J . In (3.83), K_σ denotes a Gaussian of standard deviation σ .

3.5.2 Multi-Channel Median and PDE Evolutions

Revision upcoming 2016-02: Some statements in this section need corrections. For now, flawed statements have been deleted; a revision is being prepared.

To investigate a possible generalisation of Guichard and Morel's result to the multi-channel setting, we assume now that a smooth multi-channel image $f : \Omega \rightarrow \mathbb{R}^d$ with image domain $\Omega \subset \mathbb{R}^2$ is given, and we define the median of f within a closed domain $D \subset \Omega$ via

$$\text{med}_D f := \underset{z \in \mathbb{R}^d}{\text{argmin}} \iint_D \|f(x, y) - z\|_{(2)} \, dx \, dy. \quad (3.84)$$

We have chosen the Euclidean norm here. However, it should be noted that this is not such a strong restriction as it may appear since we are going to consider the median

over sets D that converge to a point, thereby reducing the variation of function values within D also towards zero such that different norms will yield basically the same result in the limit.

Analogous to the scalar-valued case, we can define a partial differential equation that is associated to the median filter. We say that a PDE

$$\partial_t u = F(\partial_x u, \partial_y u, \partial_{xx} u, \partial_{xy} u, \partial_{yy} u, \dots) \quad (3.85)$$

is associated to the median filter if

$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(x,y)} u - u(x,y)}{\varrho^2/6} = F(\partial_x u, \partial_y u, \partial_{xx} u, \partial_{xy} u, \partial_{yy} u, \dots), \quad (3.86)$$

where $D_\varrho(x,y)$ denotes the disk of radius ϱ around (x,y) , i.e., if

$$\text{med}_{D_\varrho(x,y)} u = u(x,y) + \frac{\varrho^2}{6} F + o(\varrho^2). \quad (3.87)$$

Note that the right-hand side F of our PDE does not depend on u itself since the shift-invariance of the median rules out such a dependency. Also, our choice of $\varrho^2/6$ as time step size corresponding to a median filter with a disk of radius ϱ as structure element is motivated by analogy to Guichard and Morel's result which will naturally reappear as a special case.

Simplification by symmetries. For simplicity, we investigate the location $x = y = 0$. Exploiting the translational and rotational invariance of the involved filtering procedures, we can assume that $(1,0)^T$ and $(0,1)^T$ are the eigenvectors for the larger and smaller eigenvalue of the structure tensor $\sum_{k=1}^d \nabla u_k \nabla u_k^T$. Since it follows then that $\langle \partial_x u, \partial_y u \rangle = 0$, we can further achieve by rotations and translations in the image range \mathbb{R}^d that $u(0,0) = 0$, $\partial_x u(0,0) = (m, 0, 0, \dots)^T$, $\partial_y u(0,0) = (0, n, 0, \dots)^T$ with $m \geq n \geq 0$.

Minimality condition. We reformulate condition (3.84) by taking partial derivatives with respect to all median entries and conclude that for the multi-channel median value $M = (M_1, \dots, M_d)^T$ the equations

$$\iint_D \frac{f_k(x,y) - M_k}{\sqrt{\sum_{j=1}^d (f_j(x,y) - M_j)^2}} dx dy \quad (3.88)$$

must hold for $k = 1, \dots, d$. Given the convexity of the median energy function, fulfilment of (3.88) is also sufficient for M to be the sought median. The drawback of using (3.88) is its singularity at M , which will require to split the integrals in the course of our calculations.

Special case. Let us clarify first the case $m \neq 0, n = 0$, in which the first derivative of the multi-channel image along the local pseudo-level line vanishes. In this case the original argument of Guichard and Morel transfers straightforward to the multi-channel situation, leading to the following result.

Revision 2016-02: Last line of (3.89) restricted to $u_{xy} = 0$.

Lemma 3.11 *Let u be analytic and*

$$\begin{aligned} u(0,0) = 0, \quad \partial_x u(0,0) = (m, 0, 0, \dots)^T, \quad \partial_y u(0,0) = (0, 0, 0, \dots)^T, \\ \partial_{yy} u(0,0) = (2\delta, 2\varepsilon, 2\zeta_1, 2\zeta_2, \dots)^T, \\ \partial_{xx} u(0,0) \text{ arbitrary, } \partial_{xy} u(0,0) = 0 \end{aligned} \quad (3.89)$$

with $|m| \geq |n| > 0$. Then we have

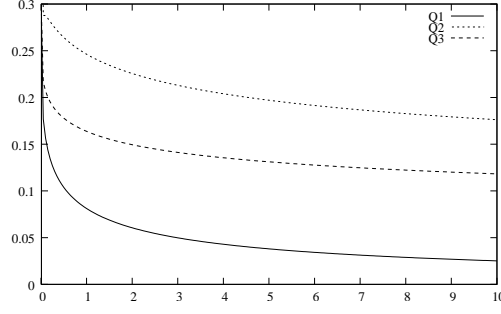
$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(0,0)} u - u(0,0)}{\varrho^2} = \frac{1}{3} \begin{pmatrix} \delta \\ \varepsilon \\ \zeta_1 \\ \zeta_2 \\ \vdots \end{pmatrix} = \frac{1}{6} \partial_{yy} u(0,0). \quad (3.90)$$

This means that in this specific case iterated median filtering approximates curvature motion even in the multi-channel setting.

Generic case. To study the situation if both first derivatives in pseudo-flow line direction and pseudo-level line direction are different from zero, we introduce now three real-valued functions which will be useful in the following. They arise as quotients of elliptic integrals.

Definition 3.6 *Let for $\lambda \in \mathbb{R} \setminus \{0\}$*

$$\begin{aligned} Q_1(\lambda) &:= \frac{\iint_{D_1} \frac{\xi^2 \eta^2}{(\xi^2 + \lambda^2 \eta^2)^{3/2}} d\xi d\eta}{\iint_{D_1} \frac{\xi^2}{(\xi^2 + \lambda^2 \eta^2)^{3/2}} d\xi d\eta}, & Q_2(\lambda) &:= \frac{\iint_{D_1} \frac{\eta^4}{(\xi^2 + \lambda^2 \eta^2)^{3/2}} d\xi d\eta}{\iint_{D_1} \frac{\eta^2}{(\xi^2 + \lambda^2 \eta^2)^{3/2}} d\xi d\eta}, \\ Q_3(\lambda) &:= \frac{\iint_{D_1} \frac{\eta^2}{(\xi^2 + \lambda^2 \eta^2)^{1/2}} d\xi d\eta}{\iint_{D_1} \frac{1}{(\xi^2 + \lambda^2 \eta^2)^{1/2}} d\xi d\eta}, \end{aligned} \quad (3.91)$$

Figure 3.17: Graphs of Q_1, Q_2, Q_3 defined in (3.91).

as well as

$$\begin{aligned} Q_1(0) = Q_2(0) = Q_3(0) &:= \frac{1}{3}, \\ Q_1(\pm\infty) = Q_2(\pm\infty) = Q_3(\pm\infty) &:= 0. \end{aligned} \quad (3.92)$$

Remark. Some of the integrals involved in these definitions cannot be resolved in closed form, while the remaining ones reduce to elliptic integrals. We do not follow this line further since for our qualitative discussion it is more valuable to retain the structural analogy between all cases. We remark that Q_1, Q_2 and Q_3 are even functions:

$$Q_1(-\lambda) = Q_1(\lambda), \quad Q_2(-\lambda) = Q_2(\lambda), \quad Q_3(-\lambda) = Q_3(\lambda). \quad (3.93)$$

We start now by considering important special cases.

Lemma 3.12 *Let u be analytic and*

$$\begin{aligned} u(0,0) = 0, \quad \partial_x u(0,0) = (m, 0, 0, \dots)^T, \quad \partial_y u(0,0) = (0, n, 0, \dots)^T, \\ \partial_{xx} u(0,0) = (2\alpha, 0, 0, \dots)^T, \quad \partial_{xy} u(0,0) = \partial_{yy} u(0,0) = 0 \end{aligned} \quad (3.94)$$

with $|m| \geq |n| > 0$. Then we have

$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(0,0)} u - u(0,0)}{\varrho^2} = \begin{pmatrix} a_0 \\ 0 \\ \vdots \end{pmatrix}, \quad (3.95)$$

where

$$a_0 = \frac{\iint_{D_1} \xi^2 \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} d\xi d\eta}{\iint_{D_1} \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} d\xi d\eta} \cdot \alpha = Q_1\left(\frac{m}{n}\right) \alpha. \quad (3.96)$$

Proof. In the given case, u is approximated around the origin by

$$u(x, y) = \begin{pmatrix} mx + \alpha x^2 \\ ny \\ 0 \\ \vdots \end{pmatrix} + O(x^2 + y^2). \quad (3.97)$$

Clearly, only the first two dimensions of \mathbb{R}^d are relevant in this case, so we will assume $d = 2$. Assuming that

$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(0,0)} u - u(0,0)}{\varrho^2} = \begin{pmatrix} a(\varrho) \\ b(\varrho) \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \quad (3.98)$$

we obtain from (3.88) the conditions

$$\begin{aligned} 0 &= \iint_{D_\varrho} \frac{mx + \alpha x^2 - a\varrho^2}{\sqrt{(mx + (\alpha x^2 - a)\varrho^2)^2 + (ny - b\varrho^2)^2}} dx dy \\ &= \iint_{D_\varrho} \frac{ny - b\varrho^2}{\sqrt{(mx + (\alpha x^2 - a)\varrho^2)^2 + (ny - b\varrho^2)^2}} dx dy. \end{aligned} \quad (3.99)$$

Considering the symmetry of the second equation with respect to the reflection $y \leftrightarrow -y$ and the convexity of the energy, it is evident that $b = 0$ must hold. Using this and the substitution $\xi := x/\varrho$, $\eta := y/\varrho$, we find

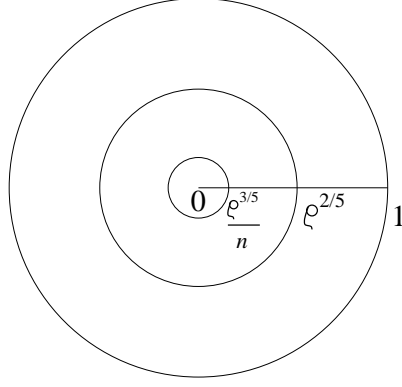
$$0 = \iint_{D_1} \frac{m\xi + (\alpha\xi^2 - a)\varrho}{\sqrt{(m\xi + (\alpha\xi^2 - a)\varrho)^2 + (n\eta)^2}} d\xi d\eta. \quad (3.100)$$

Assuming $\varrho \ll 1$, we split the integration domain D_1 into an inner disk of radius $\varrho^{3/5}/n$, an annulus with inner radius $\varrho^{3/5}/n$ and outer radius $\varrho^{2/5}$ and the remaining annulus.

1. For the inner disk we estimate

$$\iint_{D_{\varrho^{3/5}/n}} \frac{m\xi + (\alpha\xi^2 - a)\varrho}{\sqrt{(m\xi + (\alpha\xi^2 - a)\varrho)^2 + (n\eta)^2}} d\xi d\eta = O(\varrho^{6/5}) \quad (3.101)$$

since the absolute value of the integrand is bounded by 1.

Figure 3.18: Partition of the integration domain D_1 into three concentric regions.

2. For the middle annulus we find

$$\begin{aligned}
& \iint_{D_{\rho^{2/5}} \setminus D_{\rho^{3/5}/n}} \frac{m\xi + (\alpha\xi^2 - a)\rho}{\sqrt{(m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} d\xi d\eta \\
&= \iint_{(D_{\rho^{2/5}} \setminus D_{\rho^{3/5}/n}) \cap (\mathbb{R}_0^+ \times \mathbb{R})} \left(\frac{m\xi + (\alpha\xi^2 - a)\rho}{\sqrt{(m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} \right. \\
&\quad \left. + \frac{-m\xi + (\alpha\xi^2 - a)\rho}{\sqrt{(-m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} \right) d\xi d\eta \\
&= O(\rho^{6/5}) \tag{3.102}
\end{aligned}$$

since the integrand is bounded by $O(\rho^{2/5})$ according to

$$\begin{aligned}
& \frac{m\xi + (\alpha\xi^2 - a)\rho}{\sqrt{(m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} + \frac{-m\xi + (\alpha\xi^2 - a)\rho}{\sqrt{(-m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} \\
&= m\xi \left(\frac{1}{\sqrt{(m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} - \frac{1}{\sqrt{(-m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} \right) \\
&\quad + (d\xi^2 - a)\rho \left(\frac{1}{\sqrt{(m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} + \frac{1}{\sqrt{(-m\xi + (\alpha\xi^2 - a)\rho)^2 + (n\eta)^2}} \right) \\
&= \underbrace{\left(\frac{1}{\sqrt{\left(1 + (\alpha\xi^2 - a)\frac{\rho}{m\xi}\right)^2 + \left(\frac{n\eta}{m\xi}\right)^2}} - \frac{1}{\sqrt{\left(1 - (\alpha\xi^2 - a)\frac{\rho}{m\xi}\right)^2 + \left(\frac{n\eta}{m\xi}\right)^2}} \right)}_{O(\rho^{2/5})} \\
&\quad + (d\xi^2 - a) \underbrace{\frac{\rho}{m\xi}}_{O(\rho^{2/5})} \underbrace{\left(\frac{1}{\sqrt{\left(1 + (\alpha\xi^2 - a)\frac{\rho}{m\xi}\right)^2 + \left(\frac{n\eta}{m\xi}\right)^2}} + \frac{1}{\sqrt{\left(1 - (\alpha\xi^2 - a)\frac{\rho}{m\xi}\right)^2 + \left(\frac{n\eta}{m\xi}\right)^2}} \right)}_{O(1)} \\
&= O(\rho^{2/5}), \tag{3.103}
\end{aligned}$$

while the integration domain has volume $O(\rho^{4/5})$.

3. For the outer annulus we use the expansion

$$(p+q)^{-1/2} = p^{-1/2} \left(1 - \frac{q}{2p} + \mathcal{O}\left(\frac{q^2}{p^2}\right) \right) \quad (3.104)$$

to obtain

$$\begin{aligned} & \iint_{D_1 \setminus D_{\varrho^{2/5}}} \frac{m\xi + (\alpha\xi^2 - a)\varrho}{\sqrt{(m\xi + (\alpha\xi^2 - a)\varrho)^2 + (n\eta)^2}} d\xi d\eta \\ &= \underbrace{\iint_{D_1 \setminus D_{\varrho^{2/5}}} \frac{m\xi}{\sqrt{(m\xi)^2 + (n\eta)^2}} d\xi d\eta}_{=0} \\ &+ \iint_{D_1 \setminus D_{\varrho^{2/5}}} \frac{(\alpha\xi^2 - a)\varrho}{((m\xi)^2 + (n\eta)^2)^{3/2}} ((m\xi)^2 + (n\eta)^2 - (m\xi)^2) d\xi d\eta + \mathcal{O}(\varrho^{6/5}) \\ &= \iint_{D_1 \setminus D_{\varrho^{2/5}}} \varrho \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} (\alpha\xi^2 - a) d\xi d\eta + \mathcal{O}(\varrho^{6/5}). \end{aligned} \quad (3.105)$$

4. Finally, we have that

$$\begin{aligned} & \iint_{D_{\varrho^{2/5}}} \varrho \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} (\alpha\xi^2 - a) d\xi d\eta \\ &= \iint_{D_{\varrho^{2/5}}} \varrho \underbrace{\frac{(n\eta)^2}{(m\xi)^2 + (n\eta)^2}}_{\mathcal{O}(1)} \cdot \frac{1}{\sqrt{(m\xi)^2 + (n\eta)^2}} (\alpha\xi^2 - a) d\xi d\eta \\ &= \int_0^{\varrho^{2/5}} \int_0^{2\pi} \mathcal{O}(\varrho) \mathcal{O}(r^{-1}) \cdot r d\varphi dr \\ &= \mathcal{O}(\varrho) \cdot \int_0^{\varrho^{2/5}} dr = \mathcal{O}(\varrho^{7/5}). \end{aligned} \quad (3.106)$$

5. Combining (3.100), (3.101), (3.102), (3.105) and (3.106) we have

$$0 = \iint_{D_1} \varrho \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} (\alpha\xi^2 - a) d\xi d\eta + \mathcal{O}(\varrho^{6/5}), \quad (3.107)$$

thus

$$a \cdot \iint_{D_1} \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} d\xi d\eta = \alpha \cdot \iint_{D_1} \xi^2 \frac{(n\eta)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} d\xi d\eta + O(\varrho^{1/5}), \quad (3.108)$$

from which the assertion follows. \square

Lemma 3.13 *Let u be analytic and*

$$\begin{aligned} u(0,0) = 0, \quad \partial_x u(0,0) = (m, 0, 0, \dots)^T, \quad \partial_y u(0,0) = (0, n, 0, \dots)^T, \\ \partial_{xx} u(0,0) = (0, 2\beta, 0, 0, \dots)^T, \quad \partial_{xy} u(0,0) = \partial_{yy} u(0,0) = 0 \end{aligned} \quad (3.109)$$

with $|m| \geq |n| > 0$. Then we have

$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(0,0)} u - u(0,0)}{\varrho^2} = \begin{pmatrix} 0 \\ b_0 \\ 0 \\ \vdots \end{pmatrix}, \quad (3.110)$$

where

$$b_0 = \frac{\iint_{D_1} \xi^2 \frac{(m\xi)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} d\xi d\eta}{\iint_{D_1} \frac{(m\xi)^2}{((m\xi)^2 + (n\eta)^2)^{3/2}} d\xi d\eta} \cdot \beta = Q_2\left(\frac{m}{n}\right)\beta. \quad (3.111)$$

Lemma 3.14 *Let u be analytic and*

$$\begin{aligned} u(0,0) = 0, \quad \partial_x u(0,0) = (m, 0, 0, \dots)^T, \quad \partial_y u(0,0) = (0, n, 0, \dots)^T, \\ \partial_{xx} u(0,0) = (0, 0, 2\gamma, 0, \dots)^T, \quad \partial_{xy} u(0,0) = \partial_{yy} u(0,0) = 0 \end{aligned} \quad (3.112)$$

with $|m| \geq |n| > 0$. Then we have

$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(0,0)} u - u(0,0)}{\varrho^2} = \begin{pmatrix} 0 \\ 0 \\ c_0 \\ 0 \\ \vdots \end{pmatrix}, \quad (3.113)$$

where

$$c_0 = \frac{\iint_{D_1} \frac{\xi^2}{((m\xi)^2 + (n\eta)^2)^{1/2}} d\xi d\eta}{\iint_{D_1} \frac{1}{((m\xi)^2 + (n\eta)^2)^{1/2}} d\xi d\eta} \cdot \gamma = Q_3\left(\frac{m}{n}\right)\gamma. \quad (3.114)$$

The proofs of Lemmas 3.13 and 3.14 are analogous to the one of Lemma 3.12.

Analogous statements are obtained for cases in which $\partial_{yy}u(0,0) \neq 0$ but $\partial_{xx}u(0,0) = \partial_{xy}u(0,0) = 0$ (note that $|m| \geq |n|$ did not enter the proof of Lemma 3.12).

Since the above considerations were based in all cases on linearisations, the contributions of the different second derivative components superpose linearly. Thus we have the following statement.

Revision 2016-02: Last line of (3.89) restricted to $u_{xy} = 0$. Corrected version of this proposition without the restriction being prepared.

Proposition 3.15 *Let u be analytic and*

$$\begin{aligned} u(0,0) = 0, \quad \partial_x u(0,0) = (m, 0, 0, \dots)^T, \quad \partial_y u(0,0) = (0, n, 0, \dots)^T, \\ \partial_{xx}u(0,0) = (2\alpha, 2\beta, 2\gamma_1, 2\gamma_2, \dots)^T, \quad \partial_{yy}u(0,0) = (2\delta, 2\varepsilon, 2\zeta_1, 2\zeta_2, \dots)^T, \quad (3.115) \\ \partial_{xy}u(0,0) = 0 \end{aligned}$$

with $|m| \geq |n| > 0$. Then we have

$$\lim_{\varrho \rightarrow +0} \frac{\text{med}_{D_\varrho(0,0)} u - u(0,0)}{\varrho^2} = \begin{pmatrix} Q_1(m/n)\alpha + Q_2(n/m)\delta \\ Q_2(m/n)\beta + Q_1(n/m)\varepsilon \\ Q_3(m/n)\gamma_1 + Q_3(n/m)\zeta_1 \\ Q_3(m/n)\gamma_2 + Q_3(n/m)\zeta_2 \\ \vdots \end{pmatrix}, \quad (3.116)$$

where $Q_1(\lambda)$, $Q_2(\lambda)$, $Q_3(\lambda)$ are defined as in (3.91).

Transforming back to the general situation leads to the following conclusion.

Corollary 3.16 *Revision 2016-02: Corollary removed for correction.*

To summarise, we have seen that only if the first derivative of the multi-channel image in pseudo-level line direction vanishes, the multi-channel median filter provides an approximation to the curvature motion equation, see Lemma 3.11. The corollary reveals, however, that on transition to the generic case $m \neq 0$, $n \neq 0$ one obtains not only a mixture of the second derivatives in pseudo-level line direction v and pseudo-flow line direction z , but the influence of $\partial_{vv}u$ is subject to different decay characteristics for its components parallel to $\partial_v u$, $\partial_z u$ and perpendicular to both, as described by $Q_1(n/m)$, $Q_2(n/m)$, and $Q_3(n/m)$. An analogous decomposition applies to $\partial_{zz}u$.

The different decay characteristics preclude a comparably simple PDE approximation property for the multi-channel median filter as in the scalar-valued case.

Part III

Dynamical Systems in Image Processing

Chapter 4

Locally Analytic and Semi-Analytic Schemes for Diffusion Filters

In this and the following chapter, we are concerned with spatial discretisations of time-parametrised partial differential equation processes that are used as image filters. Spatial discretisation of a PDE on a grid leads naturally to a dynamical system of ordinary differential equations, with one variable for each pixel. This time-continuous dynamical system turns further into a time-discrete dynamical system when the PDE process is finally fully discretised.

Nevertheless, the time-continuous dynamical system representing the semi-discrete (spatially discretised, time-continuous) filtering process lends itself as an excellent object of study for several reasons. First, the ODE system mirrors many important properties of the fully continuous process. Second, ODEs are simple enough to allow substantial theoretical analysis, the potential of which has by far not been exhausted up to now. Third, in image processing the spatial discretisation is often determined by the image acquisition process and not dispoible, while time discretisation is a technical aspect of the image processing algorithm and within full control of algorithmical design. In many application contexts therefore a time-continuous analysis reveals the limits of what can be achieved given the available space-discrete data.

Another interesting aspect relates to the outstanding role of discontinuities in image processing. Since discontinuities in image data often represent essential image features, such as edges, they must not be treated as perturbations that are regularised away by numerical means. On the other hand, discontinuities are difficult to handle in partial differential equations. In contrast, ordinary differential equations with discontinuities can be analysed in important cases including those relevant for the image

filters we want to study. There exists a well-developed theory [90] which we won't invoke in full extent in this work; instead, we will develop some simple notions here which can be seen as special cases of concepts in [90].

Our semidiscrete dynamical system approach sketched above can be exploited in several directions. In this chapter we focus on diffusion processes, including singular ones, whose continuous formulation has been introduced in Chapter 2. We devise and analyse an interesting class of numerical methods based on a splitting of the dynamical system into local neighbourhoods – cells just big enough to be capable of representing the essential features of the diffusion process. The resulting schemes combine favourable stability properties with algorithmic simplicity. Although these advantages are bought at the price of having only conditional consistency, the numerical dissipation is extraordinarily low such that on realistic data the accuracy of computations, in particular the representation of fine details, is excellent and at least competitive with some of the best pre-existing algorithms in this field.

At the same time, the minimalistic signals and images that are used as building blocks for our numerical schemes stand in a close relation to Haar wavelets. Therefore, our schemes can immediately be interpreted as Haar wavelet shrinkage on a single scale with specific shrinkage rules [191, 221, 227, 228], which in the 1D case also inspired the actual development of the LAS. Extending these shrinkage rules to a multiscale wavelet setting leads also to novel multiscale image filters [221], which we do not describe here.

We will start by an introductory discussion of the 1D case of total variation diffusion. Many of the main ideas and design principles can be seen already in this simple situation, some in even more perfect form (like equivalence instead of approximation in some places). Then we develop the framework for isotropic nonlinear diffusion in 2D as formulated in 2.2, followed by its extension to anisotropic nonlinear diffusion in 2D, compare 2.3. Modifications for multi-channel images as well as higher dimensional image domains will be discussed at the different stages of the exposition.

While the 1D considerations in this chapter are mostly based on [191, 146], the 2D case has been treated in [221, 227, 228]. Among the co-authors of these papers, the author wants to acknowledge particularly the work of G. Steidl and P. Mrázek on the connections to wavelet methods. T. Brox and G. Steidl have substantially contributed to the equivalence result for TV flow and TV regularisation in 1D.

Related work. The idea behind locally analytic schemes can be traced back at least to the work of Richardson, Ferrell and Long [171] in computational fluid dynamics. They use, however, multiplicative splittings, which does not enable fully shift-invariant schemes for nonlinear processes. Another approach which uses analytic solutions for simple localised sub-systems to construct numerical schemes for PDE-

based image filters are short-time kernels, as in [189] where local linear diffusion processes solved by Gaussian convolution are used to approximate nonlinear diffusion.

Our analysis of the dynamical system for space-discrete singular diffusion is also related to the work of Pollak et al. [164, 165] on stabilised inverse diffusion equation (SIDEs). Analytic considerations for a version of Perona–Malik diffusion have been made by Esedoglu [83].

The locally (semi-)analytic schemes discussed here are alternatives to other numerical schemes for the same diffusion processes. In the case of singular diffusion equations, we mention here the finite difference schemes in [37], finite element schemes in [88], and a level set approach in [73]. In the context of tensor-driven anisotropic diffusion, early numeric approaches can be found in [112, 167, 206]. Later on, specific efforts to achieve good rotational invariance and to suppress numerical dissipation have been made by Weickert and Scharf [214] using Sobel-like derivative approximations, or Wang [203] with Simoncelli’s derivative approximations [188].

As mentioned above, our schemes are linked to wavelet shrinkage methods [77], particularly their shift-invariant [68] versions and those with improved rotational invariance in 2D [121]. In the anisotropic situation, there are also wavelet families designed specifically for the representation and processing of oriented structures, like contourlets [74], ridgelets [76], curvelets [50, 51], and shearlets [128].

There also exists a rich literature on connections between wavelet shrinkage and PDE methods, mostly in the continuous setting [11, 35, 56, 57, 66, 67, 142, 184, 185] but also in the discrete situation [70]. Many authors have also devised algorithms combining PDE and wavelet ideas [52, 61, 69, 80, 81, 133, 136, 137].

4.1 Two-Pixel Approach to 1D Singular Differential Equations

Most of the ideas developed in this chapter originated from the analysis of discrete 1D signals. We therefore start our considerations with a prototypic 1D case, namely the class of singular nonlinear diffusion equations with diffusivities given by (2.26).

4.1.1 Discretisation of 1D Total Variation Diffusion

We are concerned here with the 1D total variation (TV) diffusion equation

$$\partial_t u = \partial_x \left(\frac{u_x}{|u_x|} \right) \quad (4.1)$$

for the function $u : \Omega \times \mathbb{R}_0^+$ that depends on the 1D spatial variable $x \in \Omega = [a, b] \subset \mathbb{R}$ and the time variable $t \in [0, \infty)$, with the initial condition $u(x, 0) = f(x)$ and Neumann boundary conditions $\partial_x u(a) = \partial_x u(b) = 0$.

Spatial discretisation. We want to discretise (4.1) in space on a 1D regular grid with mesh size $h = 1$ while retaining the continuous time variable. Without loss of generality, we assume that the grid consists of the locations $1, 2, \dots, n$, and we denote the function values at these locations by u_1, \dots, u_n . Adapting to the terminology of (2D) image processing, we will sometimes use the term “pixel” also to refer to the samples of a 1D signal.

For the moment, we leave aside the problem of locations with $u_x = 0$. Discretising first the outer derivative by a central difference, we have then

$$\left(\partial_x \left(\frac{u_x}{|u_x|} \right) \right)_i \approx \left(\frac{u_x}{|u_x|} \right)_{i+1/2} - \left(\frac{u_x}{|u_x|} \right)_{i-1/2}. \quad (4.2)$$

On the other hand, u_x can be discretised at the inter-pixel position $i + 1/2$ by a central difference, too, yielding

$$(u_x)_{i+1/2} \approx u_{i+1} - u_i \quad (4.3)$$

where the right-hand side again consists of values at grid locations.

Since $u_x/|u_x|$ is basically the sign of u_x , we can combine everything into the following discretisation of (4.1) for an inner pixel i :

$$\partial_t u_i = \left(\partial_x \left(\frac{u_x}{|u_x|} \right) \right)_i \approx \operatorname{sgn}(u_{i+1} - u_i) - \operatorname{sgn}(u_i - u_{i-1}). \quad (4.4)$$

The latter expression is also applicable when u_x vanishes. The exact treatment of this case will become evident in the next paragraph, see (4.6).

Neumann boundary conditions can be realised by setting $u_0 := u_1$ and $u_{n+1} := u_n$ and using equation (4.4) also for $i = 1$ and $i = n$. We have therefore obtained the following system of equations that discretises (4.1) on the grid $\{1, \dots, n\}$:

$$\begin{aligned} \dot{u}_1 &= \operatorname{sgn}(u_2 - u_1) \\ \dot{u}_i &= \operatorname{sgn}(u_{i+1} - u_i) - \operatorname{sgn}(u_i - u_{i-1}), \quad i = 2, \dots, n-1, \\ \dot{u}_n &= -\operatorname{sgn}(u_n - u_{n-1}). \end{aligned} \quad (4.5)$$

Concept of solution. The system (4.5) is a dynamical system of ordinary differential equations with discontinuous right-hand sides. Before we can proceed, we have therefore to clarify what functions will be considered as *solutions* of this system.

We will say that a function $u = (u_1, \dots, u_n)^T$ defined on \mathbb{R}_0^+ or a subinterval with values in \mathbb{R}^n is a solution of (4.5) if the following conditions are satisfied:

- (I) The function u is absolutely continuous.
- (II) For each t in the interval of definition for which u is differentiable, it satisfies (4.5), where sgn is understood according to the following definition:

$$\begin{aligned} \operatorname{sgn}(s) &:= +1 && \text{if } s > 0, \\ \operatorname{sgn}(s) &:= -1 && \text{if } s < 0, \\ \operatorname{sgn}(s) &\in [-1, +1] && \text{if } s = 0. \end{aligned} \quad (4.6)$$

- (III) Whenever $\dot{u}_i(t)$ and $\dot{u}_{i+1}(t)$ exist for the same t , and $u_i(t) = u_{i+1}(t)$ holds, the expression $\operatorname{sgn}(u_{i+1}(t) - u_i(t))$ that appears in both the right-hand sides for $\dot{u}_i(t)$ and $\dot{u}_{i+1}(t)$ must take the same value in both expressions.

Remark. The conditions formulated here can be considered as a special case of the procedure described in Filippov's work [90] which allows to treat a more general class of ODE systems with discontinuous right-hand sides. The pixel-merging behaviour which we will encounter later on is a special case of a "sliding regime" as in [90].

Moreover, (4.6) can also be related with an interpretation of the underlying differential equation in terms of set-valued functions.

4.1.2 Total Variation Diffusion on Two Pixels

Turning back to the dynamical system (4.5), we can introduce a useful decomposition that reduces space-variant 1D total variation diffusion to a process that acts on minimalistic signals with just two pixels, and which can easily be analysed.

Decomposition of the discrete system. We see that the right-hand side of (4.5) for \dot{u}_i consists of two nonlinear contributions, each of which depends on just two neighbouring pixels: $\text{sgn}(u_{i+1} - u_i)$ and $\text{sgn}(u_i - u_{i-1})$. We can therefore consider this equation as *average* of the two equations

$$\begin{aligned}\dot{u}_i &= 2 \text{sgn}(u_{i+1} - u_i) \\ \dot{u}_i &= -2 \text{sgn}(u_i - u_{i-1}).\end{aligned}\tag{4.7}$$

Of course, the same decomposition can be carried out for \dot{u}_{i+1} :

$$\begin{aligned}\dot{u}_{i+1} &= 2 \text{sgn}(u_{i+2} - u_{i+1}) \\ \dot{u}_{i+1} &= -2 \text{sgn}(u_{i+1} - u_i).\end{aligned}\tag{4.8}$$

A two-pixel system. Combining the first equation of (4.7) and the second equation of (4.8), we have a dynamical system which involves only the two pixels u_i and u_{i+1} on both sides:

$$\begin{aligned}\dot{u}_i &= 2 \text{sgn}(u_{i+1} - u_i) \\ \dot{u}_{i+1} &= -2 \text{sgn}(u_{i+1} - u_i).\end{aligned}\tag{4.9}$$

First of all, by the isolation of two pixels this system is even simpler than (4.5). Second, it can be interpreted as discretisation of total variation diffusion analogous to (4.5) for a two-pixel signal on $\{i, i + 1\}$ with *periodic boundary conditions*: Inserting $u_{i-1} := u_{i+1}$ and $u_{i+2} := u_i$ into (4.4) gives exactly (4.9).¹

Analytic solution. System (4.9) is a dynamical system with discontinuous right-hand side involving only the two variables u_i and u_{i+1} . Keeping in mind our understanding of solution as pointed out in 4.1.1, it can be solved analytically: Setting for simplicity $i = 1$, we can rewrite the system as

$$\begin{aligned}\dot{u}_2 + \dot{u}_1 &= 0 \\ \dot{u}_2 - \dot{u}_1 &= -4 \text{sgn}(u_2 - u_1)\end{aligned}\tag{4.10}$$

and by setting $\mu := \frac{1}{2}(u_1 + u_2)$, $\delta := u_2 - u_1$ as

$$\begin{aligned}\dot{\mu} &= 0 \\ \dot{\delta} &= -4 \text{sgn} \delta.\end{aligned}\tag{4.11}$$

¹In [191], the decomposition of (4.5) into two-pixel systems was accomplished in a slightly different way by representing the equation for \dot{u}_i as a *sum*, instead of an average, of two equations. At the end, this led to the same system as (4.9) except for the factor 2 on the right-hand side. This system evolving at half the speed of the one here could then be interpreted as TV diffusion on two pixels with Neumann (reflecting) boundary conditions. The reason why we prefer the periodic setting here (as also sketched in [221]) is that it can be generalised to higher dimensions and is then in accordance with our 2D considerations later in this chapter. In 1D the two interpretations are fully exchangeable.

From the first equation we conclude immediately that μ is constant over time, while the second equation ensures firstly that the absolute value of δ decreases at a constant rate as long as $\delta \neq 0$. Secondly, as soon as δ happens to be zero, it remains zero: Assume that there is a time interval $[t_0, t_0 + \tau]$, $\tau > 0$, such that $\delta(t_0) = 0$ but $\delta(t) > 0$ for $t_0 < t \leq \tau$. It follows then that δ is differentiable in $[t_0, t_0 + \tau]$, and due to the mean value theorem of differential calculus we have that

$$\frac{\delta(t_0 + \tau) - \delta(t_0)}{\tau} = \delta'(t_0 + \vartheta\tau) \quad \text{for some } \vartheta \in (0, 1). \quad (4.12)$$

Since at $t = t_0 + \vartheta\tau$ we have $\delta(t) > 0$, it follows from (4.11) that $\delta'(t_0 + \vartheta\tau) < 0$. On the other hand, $\delta(t_0 + \tau) - \delta(t_0) > 0$ such that the sides of (4.12) have opposite signs, which is not possible.

Consequently, the only possible evolution for δ is given by

$$\delta(t) = \begin{cases} \delta(0) - 4t \operatorname{sgn}(\delta(0)), & 0 \leq t < \frac{1}{4} |\delta(0)|, \\ 0 & \text{else} \end{cases} \quad (4.13)$$

$$\delta(0) = f_2 - f_1$$

leading to the solution of (4.9):

$$u_i(t) = \begin{cases} \mu + \left(1 - \frac{4t}{|f_2 - f_1|}\right)(f_i - \mu), & 0 \leq t < \frac{1}{4} |f_2 - f_1|, \\ \mu, & t \geq \frac{1}{4} |f_2 - f_1|. \end{cases} \quad (4.14)$$

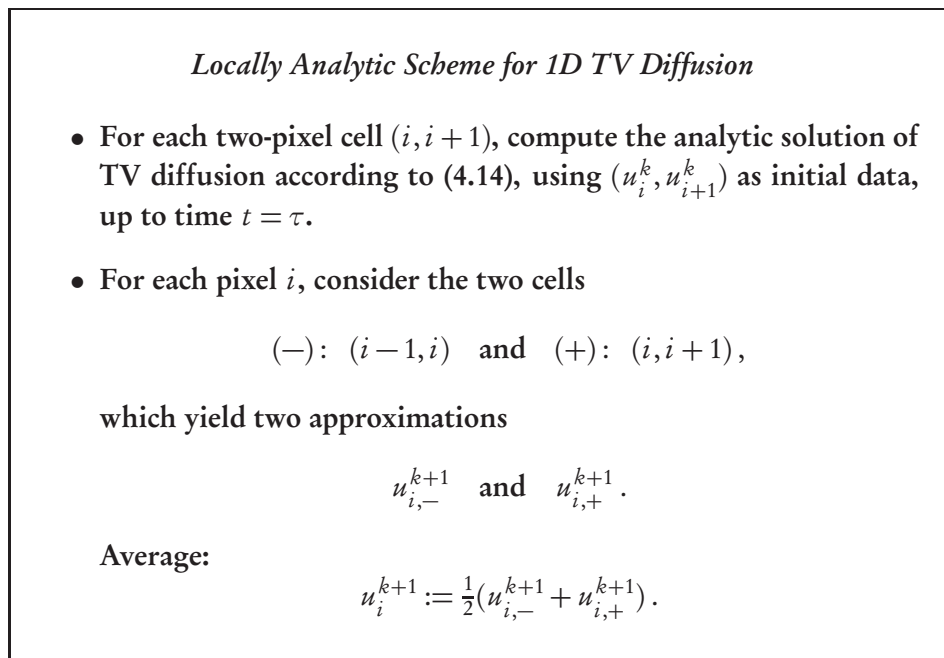


Figure 4.1: One time step of the locally analytic scheme for 1D total variation diffusion, where u^k, u^{k+1} refer to the old and new time step, resp.

4.1.3 A Numerical Scheme for 1D Total Variation Diffusion

The analytic solution (4.14) can be used to construct a numerical scheme for TV diffusion in 1D. To this end, the decomposition of the N -pixel dynamical system (4.5) into two-pixel systems of type (4.9) is used, and in each time step the analytic solution is used in each two-pixel cell to compute an approximation to the TV flow. The resulting numerical scheme reads

$$u_i^{k+1} = u_i^k + \tau \operatorname{sgn}(u_{i-1}^k - u_i^k) \min \left\{ 1, \frac{1}{4\tau} |u_i^k - u_{i-1}^k| \right\} + \tau \operatorname{sgn}(u_{i+1}^k - u_i^k) \min \left\{ 1, \frac{1}{4\tau} |u_{i+1}^k - u_i^k| \right\} \quad (4.15)$$

One time step of the algorithm is depicted in Figure 4.1. We will also refer to this scheme as *two-pixel scheme*. Since it uses analytic solutions on small signal patches, we speak also of a *locally analytic scheme*.

Stability. With the abbreviations

$$m_1 := \min \left\{ 1, \frac{4\tau}{|u_i^k - u_{i-1}^k|} \right\}, \quad m_2 := \min \left\{ 1, \frac{4\tau}{|u_{i+1}^k - u_i^k|} \right\} \quad (4.16)$$

one can rewrite (4.15) as

$$u_i^{k+1} = m_1 u_{i-1}^k + (1 - m_1 - m_2) u_i^k + m_2 u_{i+1}^k. \quad (4.17)$$

As one easily checks, $m_1, m_2 \in [0, 1/4]$ holds for arbitrary τ such that the right-hand side of (4.17) is a convex combination of $u_{i-1}^k, u_i^k, u_{i+1}^k$. Thus, our two-pixel scheme obeys the maximum-minimum principle and is absolutely stable for each τ .

Consistency analysis. To investigate the consistency of (4.15), we have to admit an arbitrary spatial mesh size h instead of fixing it to 1. The adapted version of (4.15) reads

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \frac{1}{h} \operatorname{sgn}(u_{i-1}^k - u_i^k) \min \left\{ 1, \frac{1}{4\tau} |u_i^k - u_{i-1}^k| \right\} + \frac{1}{h} \operatorname{sgn}(u_{i+1}^k - u_i^k) \min \left\{ 1, \frac{1}{4\tau} |u_{i+1}^k - u_i^k| \right\}. \quad (4.18)$$

Provided that

$$\tau \leq \frac{h}{4} \min \{ |u_{i+1}^k - u_i^k|, |u_i^k - u_{i-1}^k| \}, \quad (4.19)$$

both minimum expressions take the value 1. Then (4.18) is equivalent to the simple explicit scheme

$$u_i^{k+1} = u_i^k + \frac{\tau}{h} \operatorname{sgn}(u_{i+1}^k - u_i^k) - \frac{\tau}{h} \operatorname{sgn}(u_i^k - u_{i-1}^k) \quad (4.20)$$

and therefore an $O(\tau + b^2)$ -approximation to TV flow. For larger τ it is easy to see that linear diffusion $\partial_t u = \Delta u$ is approximated. We have therefore conditional consistency with the condition (4.19).

The inconsistency that is encountered when the condition is violated appears in those regions of the signal which are already nearly constant. In these regions the difference between linear diffusion and TV flow is small, though.

Conditional consistency is a characteristic property that can be observed for absolutely stable explicit schemes. A typical example for this behaviour is the DuFort–Frankel method [79].

4.1.4 Other Singular Nonlinear Diffusion Processes in 1D

The preceding investigations can easily be extended to the larger class of singular nonlinear diffusion processes with diffusivity of type (2.26). In 1D such a diffusion process for arbitrary $p \in [1, \infty)$ is given by the PDE

$$\partial_t u = \operatorname{div} \left(\frac{u_x}{|u_x|^p} \right). \quad (4.21)$$

This includes among others the case $p = 2$ which corresponds to balanced forward-backward diffusion in higher dimensions.²

Discretisation and decomposition into two-pixel systems can be carried out in full analogy to the TV case. The dynamical system that replaces (4.9) reads

$$\begin{aligned} \dot{u}_i &= \frac{2 \operatorname{sgn}(u_{i+1} - u_i)}{|u_{i+1} - u_i|^{p-1}} \\ \dot{u}_{i+1} &= -\frac{2 \operatorname{sgn}(u_{i+1} - u_i)}{|u_{i+1} - u_i|^{p-1}}, \end{aligned} \quad (4.22)$$

and in analogy to (4.14) its analytical solution for $i = 1$ is

$$u_i(t) = \begin{cases} \mu + \left(1 - \frac{4pt}{|f_2 - f_1|^p}\right)^{1/p} (f_i - \mu), & 0 \leq t < \frac{1}{4p} |f_2 - f_1|^p, \\ \mu, & t \geq \frac{1}{4p} |f_2 - f_1|^p. \end{cases} \quad (4.23)$$

By plugging in this solution instead of (4.14) into the algorithm depicted in Figure 4.1 we obtain a locally analytic scheme for singular nonlinear 1D diffusion with diffusivity (4.21).

²The name *balanced forward-backward diffusion* seems however inadequate in the 1D case because the forward diffusion component in (2.31) disappears completely.

4.1.5 Analytic Solution of 1D Total Variation Diffusion on n pixels

For the following considerations we return to the case of total variation diffusion, i.e., $p = 1$. The 1D situation is particularly favourable in that it is possible to solve analytically even the system (4.5) that describes spatially discrete 1D total variation flow on the n -pixel signal (u_1, \dots, u_n) . This result does not only allow in principle a direct computation of TV flow results on signals of arbitrary length. It will also lead us to an important theoretical insight.

Proposition 4.1 *The system (4.5) with initial condition $u(0) = f$ has a unique solution $u(t)$ in the sense of the conditions (I)–(III) from 4.1.1. This solution is characterised by the following properties:*

(i) **Finite extinction time.** *There exists a finite time $T \geq 0$ such that for all $t \geq T$ the signal is constant:*

$$u_i(t) = \frac{1}{n} \sum_{j=1}^n f_j \quad \text{for } i = 1, \dots, n. \quad (4.24)$$

(ii) **Finite number of merging events.** *There exists a finite sequence $0 = t_0 < t_1 < \dots < t_e = T$ such that for each $j \in \{0, 1, \dots, e-1\}$ and $i \in \{1, \dots, n-1\}$ one has either*

$$u_i(t) = u_{i+1}(t) \quad \text{for all } t \in [t_j, t_{j+1}) \quad (4.25)$$

or

$$u_i(t) \neq u_{i+1}(t) \quad \text{for all } t \in [t_j, t_{j+1}). \quad (4.26)$$

(iii) **Non-increasing contrast.** *For given $i \in \{1, \dots, n-1\}$ and $j \in \{0, 1, \dots, e-1\}$ the absolute difference $|u_{i+1}(t) - u_i(t)|$ is non-increasing throughout $[t_j, t_{j+1})$.*

(iv) **Analytical solution.** *Within each subinterval $[t_j, t_{j+1})$, constant regions of $u(t)$ undergo a linear evolution: For each given $i \in \{1, \dots, n\}$ there exist uniquely determined integers $l \geq 1$, $r \geq 0$ such that for all $t \in [t_j, t_{j+1})$ one has*

$$\begin{aligned} u_{i-l+1}(t) &= \dots = u_i(t) = \dots = u_{i+r}(t) \\ u_{i-l}(t) &\neq u_{i-l+1}(t) \quad \text{if } i-l \geq 1 \\ u_{i+r}(t) &\neq u_{i+r+1}(t) \quad \text{if } i+r \leq n-1. \end{aligned} \quad (4.27)$$

We call $(u_{i-l+1}, \dots, u_{i+r})$ a constant region of size $l+r$. It follows then for each $t \in [t_j, t_{j+1})$ that

$$u_i(t) = u_i(t_j) + \frac{2(t-t_j)}{l+r} s_{i,t_j}. \quad (4.28)$$

Here, we have for inner regions ($i - l \geq 1$ and $i + r \leq n - 1$)

$$s_{i,t_j} = \begin{cases} +1 & \text{if } u_i = \min\{u_{i-l}, u_i, u_{i+r+1}\} \\ -1 & \text{if } u_i = \max\{u_{i-l}, u_i, u_{i+r+1}\} \\ 0 & \text{else.} \end{cases} \quad (4.29)$$

For boundary regions that do not cover the entire signal ($i - l = 1$ or $i + r = n - 1$ but not both) we have

$$s_{i,t_j} = \begin{cases} +1/2 & \text{if } u_i < u_{i-l} \text{ or } u_i < u_{i+r+1} \\ -1/2 & \text{if } u_i > u_{i-l} \text{ or } u_i > u_{i+r+1} \end{cases} \quad (4.30)$$

and finally for $i - l = 1$ and $i + r = n - 1$ (the region is identical with the entire signal)

$$s_{i,t_j} = 0. \quad (4.31)$$

Proof. We start by proving property (ii). Because of requirement (I), u is almost everywhere differentiable. Let us assume that there is an interval $[\hat{t}_0, \hat{t}_1]$ such that u is differentiable for all $t \in (\hat{t}_0, \hat{t}_1)$, and there are two pixels $i, i + 1$ such that

$$\begin{aligned} u_i(\hat{t}_0) &= u_{i+1}(\hat{t}_0) \\ u_i(t) &< u_{i+1}(t) \quad \text{for } \hat{t}_0 < t \leq \hat{t}_1. \end{aligned} \quad (4.32)$$

We let $v(t) := u_{i+1}(t) - u_i(t)$. Clearly, v is a differentiable function in $[\hat{t}_0, \hat{t}_1]$ with $v(\hat{t}_1) > v(\hat{t}_0)$. By the mean value theorem there exists some $t^* \in (\hat{t}_0, \hat{t}_1)$ such that

$$v'(t^*) = \frac{v(\hat{t}_1) - v(\hat{t}_0)}{\hat{t}_1 - \hat{t}_0} > 0. \quad (4.33)$$

On the other hand, (4.5) implies

$$\begin{aligned} v'(t^*) &= \text{sgn}(u_{i+2}(t^*) - u_{i+1}(t^*)) - 2 \text{sgn}(u_{i+1}(t^*) - u_i(t^*)) \\ &\quad + \text{sgn}(u_i(t^*) - u_{i-1}(t^*)) \end{aligned} \quad (4.34)$$

where $\text{sgn}(u_{i+1}(t^*) - u_i(t^*)) = 1$ and, according to requirement (I), $\text{sgn}(u_{i+2}(t^*) - u_{i+1}(t^*)) \leq 1$ and $\text{sgn}(u_i(t^*) - u_{i-1}(t^*)) \leq 1$. Combining these equalities and inequalities, we conclude that

$$v'(t^*) \leq 0 \quad (4.35)$$

contradicting (4.33).

Analog conclusions hold if $u_i(t) > u_{i+1}(t)$ for $t > \hat{t}_0$ is assumed. Together with the finite number of pixels in our signal, (ii) follows.

Assertion (iii) follows from a slight variation of the previous argument. Let some time t be given at which u is differentiable. From (4.34) it follows for any two pixels $i, i + 1$ with $u_i(t) < u_{i+1}(t)$ that $\dot{u}_{i+1}(t) - \dot{u}_i(t) \leq 0$.

Addressing (iv), the existence of the integers l, r defining the constant region around pixel i is clear from the preceding. It remains to show the evolution given by (4.28)–(4.31). To this end, we set

$$s_+ := \begin{cases} +1 & \text{if } i+r \leq n-1 \text{ and } u_{i+r} < u_{i+r+1} \\ -1 & \text{if } i+r \leq n-1 \text{ and } u_{i+r} > u_{i+r+1} \\ 0 & \text{else (i.e., if } i+r = n-1) \end{cases} \quad (4.36)$$

and analogously

$$s_- := \begin{cases} +1 & \text{if } i+r \leq n-1 \text{ and } u_{i-l} < u_{i-l+1} \\ -1 & \text{if } i+r \leq n-1 \text{ and } u_{i-l} > u_{i-l+1} \\ 0 & \text{else (i.e., if } i-l = 0). \end{cases} \quad (4.37)$$

Summing over the equations from (4.5) with left-hand sides $\dot{u}_{i-l+1}, \dots, \dot{u}_{i+r}$ and taking into account requirement (III) from 4.1.1 we obtain

$$\dot{u}_{i-l+1}(t) + \dots + \dot{u}_i + \dots + \dot{u}_{i+r}(t) = s_+ - s_- . \quad (4.38)$$

Since we have $u_{i-l+1} = \dots = u_i = \dots = u_{i+r}$ throughout the interval $[t_j, t_{j+1})$, it follows that we have also $\dot{u}_{i-l+1}(t) = \dots = \dot{u}_i(t) = \dots = \dot{u}_{i+r}(t)$, thus

$$(l+r)\dot{u}_i = s_+ - s_- . \quad (4.39)$$

One easily verifies that in all cases, $s_+ - s_- = a_{i,t_j}$. Division of (4.39) by $(l+r)$ and integration over t gives (4.28)–(4.31).

Finally, we show (i). Whenever at some time $t > 0$ two pixels have different values, the linear evolution (4.28)–(4.31) hold. They ensure that after finite time some pixels in the signal merge, i.e., we have indeed $t \in [t_j, t_{j+1})$ where the times t_j, t_{j+1} correspond to merging events (or $t_j = 0$). Since there are only finitely many pixels in the signal, this process ends only when all pixels are merged into one constant region, whose value then remains constant over time due to (4.28), (4.31).

Summation over all equations from (4.5) finally ensures that the average grey value $\frac{1}{n} \sum_{i=1}^n u_i(t) = \frac{1}{n} \sum_{i=1}^n f_i$ is conserved throughout the evolution. In the constant final state, all pixels must therefore take exactly this value. \square

Equivalence to TV regularisation. Like TV flow, also TV regularisation can be solved analytically in the 1D spatially discrete situation. We will now sketch this solution, too, and then compare it to our previously obtained 1D TV flow result.

Total variation regularisation (compare (2.29)) of a 1D signal $f = (f_1, \dots, f_n) \in \mathbb{R}^n$ means to find a signal $u = (u_1, \dots, u_n) \in \mathbb{R}^n$ that minimises the function

$$E(u) := \sum_{i=1}^n (u_i - f_i)^2 + 2\alpha \sum_{i=1}^{n-1} |u_{i+1} - u_i| \quad (4.40)$$

with a regularisation parameter $\alpha > 0$. Here, we have assumed again Neumann boundary conditions ($u_0 = u_1$ and $u_{n+1} = u_n$).

Since (4.40) is strictly convex in \mathbb{R}^n , it has a unique minimiser. Moreover, (4.40) is a continuous function of u and α . Therefore u depends also continuously on α . The following proposition characterises this minimiser in dependence on α .

Proposition 4.2 *The energy function (4.40) has a unique minimiser $u(\alpha)$. It is characterised by the following properties:*

- (i) **Finite extinction parameter.** *There exists a finite $A \geq 0$ such that for all $\alpha \geq A$ the signal is constant:*

$$u_i(\alpha) = \frac{1}{n} \sum_{j=1}^n f_j \quad \text{for } i = 1, \dots, n. \quad (4.41)$$

- (ii) **Finite number of critical parameters.** *There exists a finite sequence $0 = \alpha_0 < \alpha_1 < \dots < \alpha_e = A$ such that for each $j \in \{0, 1, \dots, e-1\}$ and $i \in \{1, \dots, n-1\}$ one has either*

$$u_i(\alpha) = u_{i+1}(\alpha) \quad \text{for all } \alpha \in [\alpha_j, \alpha_{j+1}) \quad (4.42)$$

or

$$u_i(\alpha) \neq u_{i+1}(\alpha) \quad \text{for all } \alpha \in [\alpha_j, \alpha_{j+1}). \quad (4.43)$$

- (iii) **Non-increasing contrast.** *For given $i \in \{1, \dots, n-1\}$ and $j \in \{0, 1, \dots, e-1\}$ the absolute difference $|u_{i+1}(\alpha) - u_i(\alpha)|$ is a non-increasing function of α in the interval $[\alpha_j, \alpha_{j+1})$.*

- (iv) **Analytical solution.** *Within each subinterval $[\alpha_j, \alpha_{j+1})$, the values of constant regions of $u(\alpha)$ are linear functions of α : For each given $i \in \{1, \dots, n\}$ there exist uniquely determined integers $l \geq 1$, $r \geq 0$ such that for all $\alpha \in [\alpha_j, \alpha_{j+1})$ one has*

$$\begin{aligned} u_{i-l+1}(\alpha) &= \dots = u_i(\alpha) = \dots = u_{i+r}(\alpha) \\ u_{i-l}(\alpha) &\neq u_{i-l+1}(\alpha) && \text{if } i-l \geq 1 \\ u_{i+r}(\alpha) &\neq u_{i+r+1}(\alpha) && \text{if } i+r \leq n-1. \end{aligned} \quad (4.44)$$

We call $(u_{i-l+1}, \dots, u_{i+r})$ a constant region of size $l+r$. It follows then for each $\alpha \in [\alpha_j, \alpha_{j+1})$ that

$$u_i(\alpha) = u_i(\alpha_j) + \frac{2(\alpha - \alpha_j)}{l+r} s_{i,\alpha_j}. \quad (4.45)$$

Here, we have for inner regions ($i-l \geq 1$ and $i+r \leq n-1$)

$$s_{i,\alpha_j} = \begin{cases} +1 & \text{if } u_i = \min\{u_{i-l}, u_i, u_{i+r+1}\} \\ -1 & \text{if } u_i = \max\{u_{i-l}, u_i, u_{i+r+1}\} \\ 0 & \text{else.} \end{cases} \quad (4.46)$$

For boundary regions that do not cover the entire signal ($i-l = 1$ or $i+r = n-1$ but not both) we have

$$s_{i,\alpha_j} = \begin{cases} +1/2 & \text{if } u_i < u_{i-l} \text{ or } u_i < u_{i+r+1} \\ -1/2 & \text{if } u_i > u_{i-l} \text{ or } u_i > u_{i+r+1} \end{cases} \quad (4.47)$$

and finally for $i-l = 1$ and $i+r = n-1$ (the region is identical with the entire signal)

$$s_{i,\alpha_j} = 0. \quad (4.48)$$

This result was proven in [44] to which we also refer for this proof. An algorithm for 1D TV regularisation using similar ideas had also been developed in [165].

On comparing the two propositions, one easily detects the analogy between both results which leads to the following equivalence result.

Corollary 4.3 *In 1D, spatially discrete total variation flow (4.5) and spatially discrete total variation regularisation (4.40) of n -pixel signals are equivalent in the sense that for one and the same initial signal $f \in \mathbb{R}^n$ the solution of (4.5) at time t equals the minimiser of (4.40) for the regularisation parameter $\alpha = t$.*

4.1.6 Haar Wavelet Interpretation

The numerical scheme (4.15) is closely related to wavelet-based denoising methods known as *wavelet shrinkage* [77]. In particular, it is equivalent to shift-invariant soft Haar wavelet shrinkage [68] on a single scale. To see this, we start by sketching the procedure of Haar wavelet shrinkage in its standard form.

Haar wavelet representation of a finite signal. Let $f = (f_1, \dots, f_n)^T$ be a discrete 1D signal (in vector notation) whose length is a power of two, $n = 2^\nu$, $\nu \geq 1$. The Haar wavelet representation of f is then its representation

$$f = c_0^\nu \varphi_0^\nu + \sum_{m=1}^{\nu} \sum_{k=0}^{2^{\nu-m}-1} d_k^m \psi_k^m \quad (4.49)$$

in terms of the basis formed by the vectors

$$\begin{aligned} &\varphi_0^\nu, \psi_0^\nu, \psi_0^{\nu-1}, \psi_1^{\nu-1}, \dots, \\ &\psi_0^m, \dots, \psi_{2^{\nu-m}-1}^m, \dots, \\ &\psi_0^1, \dots, \psi_{n/2-1}^1 \end{aligned} \quad (4.50)$$

where

$$\varphi_0^\nu := \frac{1}{2^{\nu/2}}(1, \dots, 1)^T \quad (4.51)$$

is called *scaling function* and

$$\begin{aligned} \psi_k^m &:= \frac{1}{2^{m/2}}(r_0, \dots, r_{n-1})^T, \\ r_j &= \begin{cases} +1, & 0 \leq j - k \cdot 2^m \leq 2^{m-1} - 1, \\ -1, & 2^{m-1} \leq j - k \cdot 2^m \leq 2^m - 1, \\ 0, & \text{else} \end{cases} \end{aligned} \quad (4.52)$$

are called *wavelet functions*.

The coefficients c_0^ν, d_k^m can be computed by letting $c_k^0 := f_{k+1}$, $k = 0, \dots, n-1$, and using the recursion

$$c_k^{m+1} := \frac{1}{\sqrt{2}}(c_{2k}^m + c_{2k+1}^m), \quad d_k^{m+1} := \frac{1}{\sqrt{2}}(c_{2k}^m - c_{2k+1}^m) \quad (4.53)$$

for $m = 0, \dots, \nu-1$, $k = 0, \dots, 2^{\nu-m-1} - 1$. Vice versa, the recursive formulas

$$c_{2k}^m := \frac{1}{\sqrt{2}}(c_k^{m+1} + d_k^{m+1}), \quad c_{2k+1}^m := \frac{1}{\sqrt{2}}(c_k^{m+1} - d_k^{m+1}) \quad (4.54)$$

allow to recover c_k^0 , thus f_k , if only c_0^ν and all the d coefficients are given.

Haar wavelet shrinkage. To denoise a signal $f = (f_1, \dots, f_n)$, $n = 2^\nu$, one proceeds in three steps.

1. *Analysis step.* The Haar wavelet decomposition $(c_0^\nu, d_0^\nu, \dots, d_n^1)$ of the signal is computed using the recursion (4.53).

2. *Shrinkage step.* A so-called *shrinkage function* S_θ is used to modify all the wavelet coefficients,

$$\tilde{d}_k^m := S_\theta(d_k^m), \quad (4.55)$$

while c_0^ν is left unchanged.

Examples for frequently used shrinkage functions are

$$S_\theta(z) := \begin{cases} z, & |z| \geq \theta, \\ 0 & \text{else} \end{cases} \quad (\text{hard shrinkage}) \quad (4.56)$$

$$S_\theta(z) := \begin{cases} z - \text{sgn}(z)\theta, & |z| \geq \theta, \\ 0 & \text{else} \end{cases} \quad (\text{soft shrinkage}). \quad (4.57)$$

3. *Synthesis step.* The denoised signal is reconstructed from the scaling coefficient c_0^ν and the modified wavelet coefficients \tilde{d}_k^m analog to (4.54).

Cycle spinning. While the shrinkage procedure is simple and fast, it displays a severe disadvantage for the filtering of signals: It is not translation invariant. Its exact behaviour at a given pixel position depends on the place of that pixel in the dyadic hierarchy constituted by the recursive procedure (4.53).

To circumvent this problem, the *cycle spinning* procedure has been proposed. It means to perform ordinary wavelet shrinkage on the original signal of length n as well as on $n-1$ copies shifted cyclically by $1, 2, \dots, n-1$ pixels. The shrunken translates are shifted back and all N shrinkage results averaged to give the shift-invariant filtering result. Using the cyclic shift operator $C_k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ given by

$$C_k : (f_1, \dots, f_n) \mapsto (f_k + 1, \dots, f_n, f_1, \dots, f_k) \quad (4.58)$$

with inverse $C_k^{-1} = C_{n-k}$ and abbreviating the complete shrinkage operation (including analysis, shrinkage and synthesis step) by Σ_θ , shift-invariant shrinkage reads

$$\bar{\Sigma}_\theta(f) := \sum_{k=0}^{n-1} (C_{n-k} \circ \Sigma_\theta \circ C_k)(f). \quad (4.59)$$

In this procedure, f is treated as a periodical signal which deviates from the reflecting boundary conditions which are often used in image processing. However, if the n -pixel signal $f = (f_1, \dots, f_n)$ is simply replaced with the $2n$ -pixel signal $f' := (f_1, \dots, f_n, f_n, \dots, f_1)$, periodic boundary conditions on f' exactly simulate reflecting boundary conditions on f .

Soft Haar wavelet shrinkage on a single scale. In the simplest case, when we have a two-pixel signal (f_1, f_2) , the Haar wavelet representation boils down to a single scale,

featuring one scaling coefficient and one wavelet coefficient. We apply soft Haar wavelet shrinkage to this signal to obtain a filtered version $(\tilde{f}_1, \tilde{f}_2)$. By substituting (4.53), (4.55) with (4.57) and (4.54), we see that

$$\begin{aligned}\tilde{f}_1 &= f_1 + \max\left(\frac{\theta}{\sqrt{2}}, \frac{|f_1 - f_2|}{2}\right) \operatorname{sgn}(f_2 - f_1), \\ \tilde{f}_2 &= f_2 + \max\left(\frac{\theta}{\sqrt{2}}, \frac{|f_1 - f_2|}{2}\right) \operatorname{sgn}(f_1 - f_2).\end{aligned}\tag{4.60}$$

Comparing to (4.14), one sees that $\tilde{f}_1 = u_1(t)$ and $\tilde{f}_2 = u_2(t)$ if $t = \frac{\theta}{2\sqrt{2}}$ is used.

In the case of a larger signal of length n (for which it is now sufficient to be even), we can easily carry out a single-scale wavelet transform by evaluating (4.53) only for $m = 0$. Of course, a full representation of the signal then requires to keep all c_k^1 and all d_k^1 . Modifying the d_k^1 by the shrinkage function and reverting the transform gives a single-scale soft Haar wavelet shrinkage which is, however, still not translation invariant, as only the odd–even neighbour pixel pairs (f_{2k-1}, f_{2k}) are considered. To obtain a translation invariant filter, one uses cycle spinning. Since in the single-scale setting a shift by 2 pixels already leads to the same decomposition of the signal into pixel pairs, cycle spinning here boils down to averaging just two ordinary shrinkage results: one of the unshifted signal, and one preceded by a 1-pixel shift and followed by the inverse shift.

For each single pixel f_i this means that two filtered values $\tilde{f}_{i,+}$ and $\tilde{f}_{i,-}$ are averaged, where $\tilde{f}_{i,+}$ has been computed by shrinkage from the pixel pair (f_i, f_{i+1}) while $\tilde{f}_{i,-}$ has been obtained from the pixel pair (f_{i-1}, f_i) . (In this formulation, we are now ready to drop even the requirement that n be even.) Obviously, this is the same averaging procedure that was used to obtain the two-pixel scheme (4.15) from the analytic solutions (4.14). Taking into account the equivalence of (4.14) and (4.60), it is evident that shift-invariant soft Haar wavelet shrinkage with $\theta = 2\sqrt{2}\tau$ is equivalent to one time step of (4.15).

The link between our two-pixel scheme and wavelet shrinkage is not limited to total variation diffusion. To formulate similar results, e.g., for the more general singular diffusion schemes from 4.1.4 requires just to use a different shrinkage function. Equivalences between different shrinkage functions and diffusivities have been investigated by Mrázek et al. in [145].

4.2 Four-Pixel Approach to Isotropic Nonlinear 2D Diffusion

We will now turn from the 1D to the 2D situation which is of greater interest in the image processing context. Following common practice in image processing, we will assume throughout these considerations that the underlying spatial grid is regular with equal mesh sizes $h = 1$ in x and y direction. An exception will be made in consistency analysis where arbitrary h will be admitted. Transfer of the schemes developed here to this case is easy; however, equal mesh sizes in x and y directions are essential.

A degree of freedom that arises on transition to the 2D situation is the possibility of anisotropic diffusion, i.e., diffusion with direction-dependent diffusivity, as discussed in 2.3. We will consider this case in 4.3, while in the present section we will stick to the isotropic situation, compare 2.2.

4.2.1 Discretisation of the Divergence Form

We start by discretising the diffusion equation (2.14) in space under the premise that the diffusivity field $g = g(|\nabla u_\sigma|^2)$ is known. For abbreviation we let $v = u_\sigma$. So what is discretised in this step is in fact the *divergence form*.

We assume again that the image u is sampled on a rectangular grid of mesh width 1 with integer coordinates. As to the diffusivity g , let us assume that its values are given at the *midpoints* of cells containing 2×2 grid points of u , i.e., at the points $(i + 1/2, j + 1/2)$ with integer i, j .

First approximation. In (2.14), we find two differential operators which need to be discretised, namely the gradient ∇ and the divergence div . Expressing these by partial derivatives w.r.t. the spatial coordinates x and y and approximating these by central differences leads to

$$\begin{aligned} (\text{div}(g(|\nabla v|^2)\nabla u))_{i,j} \approx & g_{i+1/2,j} \cdot (u_{i+1,j} - u_{i,j}) - g_{i-1/2,j} \cdot (u_{i,j} - u_{i-1,j}) \\ & + g_{i,j+1/2} \cdot (u_{i,j+1} - u_{i,j}) - g_{i,j-1/2} \cdot (u_{i,j} - u_{i,j-1}). \end{aligned} \quad (4.61)$$

Since we only want to work with the diffusivities at the cell midpoints, we use linear interpolation for g and arrive at

$$\begin{aligned} (\text{div}(g(|\nabla v|^2)\nabla u))_{i,j} \approx & \frac{1}{2} \left(g_{i+1/2,j+1/2} \cdot (u_{i+1,j} + u_{i,j+1} - 2u_{i,j}) \right. \\ & + g_{i+1/2,j-1/2} \cdot (u_{i+1,j} + u_{i,j-1} - 2u_{i,j}) \\ & + g_{i-1/2,j+1/2} \cdot (u_{i-1,j} + u_{i,j+1} - 2u_{i,j}) \\ & \left. + g_{i-1/2,j-1/2} \cdot (u_{i-1,j} + u_{i,j-1} - 2u_{i,j}) \right) \end{aligned} \quad (4.62)$$

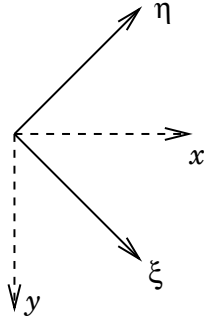


Figure 4.2: Unit vectors for standard coordinates x, y and the 45° -rotated coordinates ξ, η .

as one consistent approximation.

Second approximation. However, this is not the only possibility. Let us study now an alternative discretisation, which uses representations of div and ∇ in terms of the 45° -rotated coordinates ξ, η where

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} := H \begin{pmatrix} x \\ y \end{pmatrix}, \quad H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (4.63)$$

see Figure 4.2. Note that $H = H^T = H^{-1}$. Thereby we find

$$\begin{aligned} & (\text{div}(g(|\nabla v|^2)\nabla u))_{i,j} \\ & \approx \frac{1}{2} \left(g_{i+1/2,j+1/2} \cdot (u_{i+1,j+1} - u_{i,j}) + g_{i+1/2,j-1/2} \cdot (u_{i+1,j-1} - u_{i,j}) \right. \\ & \quad \left. + g_{i-1/2,j+1/2} \cdot (u_{i-1,j+1} - u_{i,j}) + g_{i-1/2,j-1/2} \cdot (u_{i-1,j-1} - u_{i,j}) \right) \end{aligned} \quad (4.64)$$

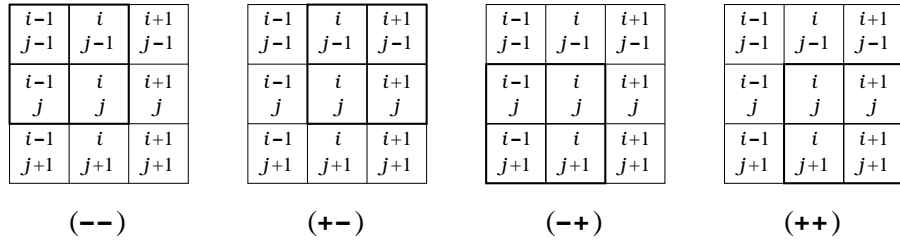
as a second discretisation.

Convex combinations. Since both (4.62) and (4.64) are consistent approximations of the right-hand side of (2.14), so are their convex combinations. For every $\alpha \in [0, 1]$ therefore the dynamical system

$$\begin{aligned} \dot{u}_{i,j} = \frac{1}{2} \left(& g_{i+1/2,j+1/2} \cdot (\alpha u_{i+1,j} + \alpha u_{i,j+1} + (1-\alpha)u_{i+1,j+1} - (1+\alpha)u_{i,j}) \right. \\ & + g_{i+1/2,j-1/2} \cdot (\alpha u_{i+1,j} + \alpha u_{i,j-1} + (1-\alpha)u_{i+1,j-1} - (1+\alpha)u_{i,j}) \\ & + g_{i-1/2,j+1/2} \cdot (\alpha u_{i-1,j} + \alpha u_{i,j+1} + (1-\alpha)u_{i-1,j+1} - (1+\alpha)u_{i,j}) \\ & \left. + g_{i-1/2,j-1/2} \cdot (\alpha u_{i-1,j} + \alpha u_{i,j-1} + (1-\alpha)u_{i-1,j-1} - (1+\alpha)u_{i,j}) \right) \end{aligned} \quad (4.65)$$

where the dot denotes differentiation with respect to t , is an approximation of (2.14).

Representation as average. An interesting observation about this dynamical system is that within each summand on the right-hand side, only quantities from one of

Figure 4.3: The four-pixel cells contributing to $u_{i,j}$.

the four-pixel cells

$$\begin{aligned}
 (---) &: \{i-1, i\} \times \{j-1, j\}, & (+-): & \{i, i+1\} \times \{j-1, j\}, \\
 (-++) &: \{i-1, i\} \times \{j, j+1\}, & (++) &: \{i, i+1\} \times \{j, j+1\},
 \end{aligned} \tag{4.66}$$

see Figure 4.3, are combined.

This enables us to rewrite (4.65) as *average* of four dynamical systems each of which contains only interactions within one cell. One such system, which we write down for simplicity for the cell $\{1, 2\} \times \{1, 2\}$, reads

$$\begin{aligned}
 \dot{u}_{1,1} &= 2g_{3/2,3/2} \cdot (-(1+\alpha)u_{1,1} + \alpha u_{1,2} + \alpha u_{2,1} + (1-\alpha)u_{2,2}), \\
 \dot{u}_{2,1} &= 2g_{3/2,3/2} \cdot (\alpha u_{1,1} + (1-\alpha)u_{1,2} - (1+\alpha)u_{2,1} + \alpha u_{2,2}), \\
 \dot{u}_{1,2} &= 2g_{3/2,3/2} \cdot (\alpha u_{1,1} - (1+\alpha)u_{1,2} + (1-\alpha)u_{2,1} + \alpha u_{2,2}), \\
 \dot{u}_{2,2} &= 2g_{3/2,3/2} \cdot ((1-\alpha)u_{1,1} + \alpha u_{1,2} + \alpha u_{2,1} - (1+\alpha)u_{2,2}).
 \end{aligned} \tag{4.67}$$

We stress that (4.67) also coincides with the application of (4.65) to a 2×2 -pixel image with *periodic* boundary conditions, because these boundary conditions surround each pixel with four identical (up to reflections) 2×2 cells. In this sense, the decomposition is in full analogy with the 1-D case.

4.2.2 Discretisation of Diffusivities

In order to apply the dynamical systems derived in the previous subsection to compute nonlinear diffusion, we need to discretise the diffusivity $g(|\nabla u_\sigma|^2)$ at cell mid-points $(i+1/2, j+1/2)$. This comes down to discretising the squared gradient magnitude $|\nabla u_\sigma|^2$, which we will again do in a twofold way.

First approximation. Abbreviating again $v = u_\sigma$, we have in standard coordinates $|\nabla v|^2 = (\partial_x v)^2 + (\partial_y v)^2$. We can approximate $\partial_x v$ and $\partial_y v$ by arithmetic means of central difference approximations,

$$\begin{aligned}
 (\partial_x v)_{i+1/2, j+1/2} &\approx \frac{1}{2}(v_{i+1, j+1} - v_{i, j+1} + v_{i+1, j} - v_{i, j}) \\
 (\partial_y v)_{i+1/2, j+1/2} &\approx \frac{1}{2}(v_{i+1, j+1} + v_{i, j+1} - v_{i+1, j} - v_{i, j})
 \end{aligned} \tag{4.68}$$

which leads to

$$(|\nabla v|^2)_{i+1/2, j+1/2} \approx \frac{1}{2}((v_{i+1, j+1} - v_{i, j})^2 + (v_{i+1, j} - v_{i, j+1})^2). \quad (4.69)$$

This expression can equally be obtained by using central difference approximations in the ξ - η coordinates to represent $|\nabla v|$.

Second approximation. Since, however, computing the gradient magnitude in fact relies on the *squares* of the partial derivatives, we can replace the averaging of central difference approximations of the derivatives themselves in (4.68) by averaging the squared derivatives, which gives

$$\begin{aligned} ((\partial_x v)^2)_{i+1/2, j+1/2} &\approx \frac{1}{2}((v_{i+1, j+1} - v_{i, j+1})^2 + (v_{i+1, j} - v_{i, j})^2) \\ ((\partial_y v)^2)_{i+1/2, j+1/2} &\approx \frac{1}{2}((v_{i+1, j+1} - v_{i+1, j})^2 + (v_{i, j+1} - v_{i, j})^2) \end{aligned} \quad (4.70)$$

and thereby a new approximation for the gradient magnitude,

$$\begin{aligned} (|\nabla v|^2)_{i+1/2, j+1/2} &\approx \frac{1}{2} \left((v_{i+1, j+1} - v_{i, j+1})^2 + (v_{i+1, j} - v_{i, j})^2 \right. \\ &\quad \left. + (v_{i+1, j+1} - v_{i+1, j})^2 + (v_{i, j+1} - v_{i, j})^2 \right). \end{aligned} \quad (4.71)$$

Each of the approximations (4.69), (4.71) and their convex combinations

$$\begin{aligned} (|\nabla v|^2)_{i+1/2, j+1/2} &\approx G_\alpha^2(v)_{i+1/2, j+1/2} \\ &:= \frac{\alpha}{2} \left((v_{i+1, j+1} - v_{i, j+1})^2 + (v_{i+1, j} - v_{i, j})^2 \right. \\ &\quad \left. + (v_{i+1, j+1} - v_{i+1, j})^2 + (v_{i, j+1} - v_{i, j})^2 \right) \\ &\quad + \frac{1-\alpha}{2} \left((v_{i+1, j+1} - v_{i, j})^2 + (v_{i+1, j} - v_{i, j+1})^2 \right), \end{aligned} \quad (4.72)$$

$\alpha \in [0, 1]$, can be used in computing the diffusivity g .

4.2.3 Variational Interpretation

In 2.2.2 we have seen that diffusion processes of type (2.14) with $\sigma = 0$, i.e., without pre-smoothing of the gradient field, can be represented as gradient descents for suitable energy functionals. We will show now that for $\sigma = 0$ our discretised diffusion processes are gradient descents for *discrete* energy functions.

We consider energy functions given by

$$E(u) := \frac{1}{2} \sum_{i, j} \Psi \left([|\nabla u|^2]_{i+1/2, j+1/2} \right) \quad (4.73)$$

where the outer sum runs over all such four-pixel cells. Here and in the following we use square brackets to denote discretisations, i.e., $[[\nabla u]^2]_{i+1/2, j+1/2}$ is some discretisation of the squared gradient magnitude $|\nabla u|^2$ within the four-pixel cell $\{i, i+1\} \times \{j, j+1\}$. As in Chapter 2, Ψ is an increasing function on \mathbb{R}_0^+ , whose derivative will show up in the gradient descent equations in the role of a diffusivity, so $\Psi' \equiv g$.

The partial derivative of the energy E with respect to some particular variable $u_{i,j}$ is computed via the chain rule:

$$\frac{\partial E}{\partial u_{i,j}} = \frac{1}{2} \sum_{i',j'} \Psi'([[\nabla u]^2]_{i'+1/2, j'+1/2}) \frac{\partial [[\nabla u]^2]_{i'+1/2, j'+1/2}}{\partial u_{i,j}}. \quad (4.74)$$

The corresponding gradient descent then is

$$\dot{u}_{i,j} = -\frac{\partial E}{\partial u_{i,j}} \quad \text{for all } i, j. \quad (4.75)$$

For given (i, j) , the sum on the right-hand side has four possibly nonzero contributions belonging to the cells (4.66), compare Fig. 4.3. When specifying a discretisation $[[\nabla u]^2]_{i'+1/2, j'+1/2}$ in (4.73), the same discretisation reappears in the argument of Ψ' in (4.75). At the same time, it determines the expression for $\partial [[\nabla u]^2]_{i'+1/2, j'+1/2} / \partial u_{i,j}$.

With the discretisation (4.72) from Subsection 4.2.2, we have

$$\begin{aligned} \frac{\partial [[\nabla u]^2]_{(\text{---})}}{\partial u_{i,j}} &= \frac{\alpha}{2} (2u_{i,j} - u_{i-1,j} - u_{i,j-1}) + \frac{1-\alpha}{2} (u_{i,j} - u_{i-1,j-1}), \\ \frac{\partial [[\nabla u]^2]_{(\text{+-})}}{\partial u_{i,j}} &= \frac{\alpha}{2} (2u_{i,j} - u_{i+1,j} - u_{i,j-1}) + \frac{1-\alpha}{2} (u_{i,j} - u_{i+1,j-1}), \\ \frac{\partial [[\nabla u]^2]_{(\text{-+})}}{\partial u_{i,j}} &= \frac{\alpha}{2} (2u_{i,j} - u_{i-1,j} - u_{i,j+1}) + \frac{1-\alpha}{2} (u_{i,j} - u_{i-1,j+1}), \\ \frac{\partial [[\nabla u]^2]_{(\text{++})}}{\partial u_{i,j}} &= \frac{\alpha}{2} (2u_{i,j} - u_{i+1,j} - u_{i,j+1}) + \frac{1-\alpha}{2} (u_{i,j} - u_{i+1,j+1}), \end{aligned} \quad (4.76)$$

where we have used the cell abbreviations from (4.66). Inserting these into (4.75), we obtain exactly (4.67) with $g = \Psi'$ as gradient descent. Note that the same value of α has to be used in (4.72) and (4.67).

4.2.4 Analytic Solutions

In investigating the dynamical system (4.67) that describes isotropic nonlinear diffusion on a 2×2 -pixel image patch, we turn first to a case in which the system can even be solved analytically.

Singular diffusivity. Let us consider the special case of the singular diffusivities (2.26), and set $\alpha = 1/2$ in the discretisations (4.65) and (4.72). By combining (4.67) and (4.72), we can then write down the following dynamical system for the four-pixel cell $\{1,2\} \times \{1,2\}$ which describes the *simultaneous* evolution of the image u and diffusivity field g :

$$\begin{aligned}\dot{u}_{1,1} &= g \cdot (-3u_{1,1} + u_{2,1} + u_{1,2} + u_{2,2}), \\ \dot{u}_{2,1} &= g \cdot (u_{1,1} - 3u_{2,1} + u_{1,2} + u_{2,2}), \\ \dot{u}_{1,2} &= g \cdot (u_{1,1} + u_{2,1} - 3u_{1,2} + u_{2,2}), \\ \dot{u}_{2,2} &= g \cdot (u_{1,1} + u_{2,1} + u_{1,2} - 3u_{2,2})\end{aligned}\tag{4.77}$$

with

$$g = g((G_{1/2}(u(t)))^2) = (G_{1/2}(u))^{-p}.\tag{4.78}$$

Let us from now on abbreviate $G_{1/2}(u)$ by $G(u)$. We observe first that for the average grey value $\mu := \frac{1}{4}(u_{1,1} + u_{2,1} + u_{1,2} + u_{2,2})$ the system (4.77) implies $\dot{\mu} = 0$, thus $\mu = \frac{1}{4}(f_{1,1} + f_{2,1} + f_{1,2} + f_{2,2})$ at all times. Furthermore (4.77) simplifies to

$$\dot{u}_{i,j} = 4g \cdot (\mu - u_{i,j}), \quad i, j = 1, 2,\tag{4.79}$$

which is again a dynamical system with discontinuous right hand side. In analogy to the concept of solution introduced for TV flow in 1D on page 103 we require also solutions of (4.79) to be absolutely continuous and satisfy the given ODEs wherever the derivatives involved are defined. Similar to the dynamical system (4.9) describing TV flow in 1D, (4.79) can then be solved analytically.

To see this, let us notice first that when all $u_{i,j} - \mu$ are replaced with $\lambda(u_{i,j} - \mu)$ with the same $\lambda > 0$, then $G(u)$ is also replaced with $\lambda G(u)$. In other words, $G(u)$ is a homogeneous function of the $u_{i,j} - \mu$ as long as their signs remain unchanged. Consider now a time interval $[0, T]$ during which the solution of (4.79) is differentiable. Provided that $G(f) \neq 0$, we can set

$$\tilde{\eta}(t) := \frac{G(u(t))}{G(f)}\tag{4.80}$$

which is then a differentiable function of t with $\tilde{\eta}(0) = 1$. Since each of the quantities $u_{i,j} - \mu$ fulfils the differential equation

$$\dot{y} = -4g(G(f)\tilde{\eta}(t)) \cdot y,\tag{4.81}$$

it follows that there is a differentiable function η with $\eta(0) = 1$ such that

$$\mu - u_{i,j}(t) = \eta(t) \cdot (\mu - f_{i,j})\tag{4.82}$$

for all i, j and $t \in [0, T]$. Due to the above mentioned homogeneity of $G(u)$ w.r.t. $u_{i,j} - \mu$, it follows that $\tilde{\eta} \equiv \eta$.

Differentiation yields $\dot{u}_{i,j} = -\dot{\eta} \cdot (\mu - f_{i,j})$ which after insertion into (4.79) leads to

$$-\dot{\eta} \cdot (\mu - f_{i,j}) = 4g \cdot (\mu - u_{i,j}) \quad (4.83)$$

and by (4.82) finally to

$$\dot{\eta} = -4g\eta. \quad (4.84)$$

From (4.79) and (4.82) it is clear that $\dot{\eta}$ and η always have opposite signs such that the absolute value of η can never increase. Consequently, since $\eta(0) = 1$, we see that $\eta(t) \geq 0$ for $t \geq 0$. With (4.80) and (2.26) it follows that

$$\dot{\eta}(t) = -4(G(f))^{-p} \eta(t)^{1-p}. \quad (4.85)$$

This equation has the solution

$$\eta(t) = \left(1 - \frac{4pt}{(G(f))^p}\right)^{1/p}, \quad (4.86)$$

in the interval $[0, T]$ within which we have a differentiable solution. Since according to (4.86) η becomes zero at $t = \frac{(G(f))^p}{4p}$, this is the greatest T that can be chosen.

An argument using the mean value theorem as in 4.1.2 makes clear, however, that once $\eta(t)$ has reached zero, and all $u_{i,j}$ have thus become equal, they remain equal forever, such that the solution can be continued on $[0, \infty)$ by $\eta(t) = 0$.

The solution for all $t \geq 0$ therefore reads

$$\eta(t) = \begin{cases} \left(1 - \frac{4pt}{(G(f))^p}\right)^{1/p}, & 0 \leq t < \frac{(G(f))^p}{4p}, \\ 0, & t \geq \frac{(G(f))^p}{4p}. \end{cases} \quad (4.87)$$

Finally, by (4.82), the analytical solution of (4.79) is given for $i, j = 1, 2$ by

$$u_{i,j}(t) = \begin{cases} \mu + \left(1 - \frac{4pt}{(G(f))^p}\right)^{1/p} (f_{i,j} - \mu), & 0 \leq t < \frac{(G(f))^p}{4p}, \\ \mu, & t \geq \frac{(G(f))^p}{4p}. \end{cases} \quad (4.88)$$

For the TV diffusion case $p = 1$, particularly, (4.88) simplifies to

$$u_{i,j}(t) = \begin{cases} \mu + \left(1 - \frac{4t}{G(f)}\right) (f_{i,j} - \mu), & 0 \leq t < \frac{G(f)}{4}, \\ \mu, & t \geq \frac{G(f)}{4}, \end{cases} \quad (4.89)$$

and shows a linear evolution which can be written in a slightly different form as

$$u_{i,j}(t) = f_{i,j} + \frac{4t}{G(f)} (\mu - f_{i,j}) \min \left\{ 1, \frac{G(f)}{4t} \right\}, \quad i, j = 1, 2. \quad (4.90)$$

For BFB diffusion, $p = 2$, we have the analytical solution

$$u_{i,j}(t) = \begin{cases} \mu + \sqrt{1 - \frac{8t}{(G(f))^2}} (f_{i,j} - \mu), & 0 \leq t < \frac{(G(f))^2}{8}, \\ \mu, & t \geq \frac{(G(f))^2}{8}. \end{cases} \quad (4.91)$$

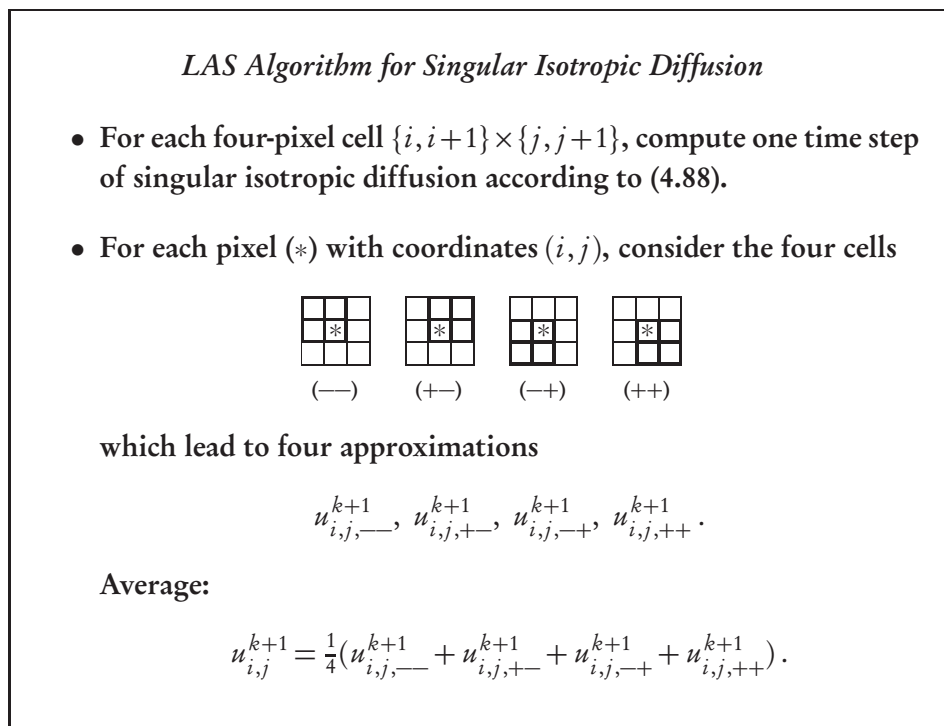


Figure 4.4: One time step of the locally analytic scheme for a singular isotropic diffusion process, where u^k, u^{k+1} refer to the old and new time step, respectively.

4.2.5 Locally Analytic Scheme for Singular Isotropic Diffusion Processes

We continue to consider the case of the singular diffusivities (2.26) with the discretisation parameter $\alpha = 1/2$. Combination of the results achieved so far enables us to construct a novel numerical scheme for singular isotropic diffusion processes.

We recall that in the case under consideration the dynamical system (4.65) is represented by the average of four dynamical systems of type (4.77). Using the notations

$$\begin{aligned}
 \mu_{i,j,-}^k &:= \frac{1}{4}(u_{i-1,j}^k + u_{i,j-1}^k + u_{i-1,j-1}^k + u_{i,j}^k), \\
 \mu_{i,j,+}^k &:= \frac{1}{4}(u_{i+1,j}^k + u_{i,j-1}^k + u_{i+1,j-1}^k + u_{i,j}^k), \\
 \mu_{i,j,-+}^k &:= \frac{1}{4}(u_{i-1,j}^k + u_{i,j+1}^k + u_{i-1,j+1}^k + u_{i,j}^k), \\
 \mu_{i,j,++}^k &:= \frac{1}{4}(u_{i+1,j}^k + u_{i,j+1}^k + u_{i+1,j+1}^k + u_{i,j}^k),
 \end{aligned} \tag{4.92}$$

we could therefore discretise (4.65) in time via an explicit Euler scheme and obtain the naive scheme

$$\begin{aligned}
 u_{i,j}^{k+1} &= u_{i,j}^k + \tau g_{i-1/2,j-1/2} \cdot (\mu_{i,j,-}^k - u_{i,j}^k) + \tau g_{i+1/2,j-1/2} \cdot (\mu_{i,j,+}^k - u_{i,j}^k) \\
 &\quad + \tau g_{i-1/2,j+1/2} \cdot (\mu_{i,j,-+}^k - u_{i,j}^k) + \tau g_{i+1/2,j+1/2} \cdot (\mu_{i,j,++}^k - u_{i,j}^k).
 \end{aligned} \tag{4.93}$$

Here τ denotes the time step size and $u^k = (u_{i,j}^k)_{i,j}$ the approximate solution at pixel (i, j) and time $k\tau$. Unfortunately, due to the singularity of g at zero, this scheme becomes unstable with respect to the maximum-minimum principle for arbitrary small time steps if neighbouring pixel values become arbitrary close. We use therefore a different approximation.

Due to the decomposition of (4.65) into the average of four 2×2 -pixel systems, its solution is approximatively the average of the solutions of the four smaller systems. By (4.88) we know these exactly. This inspires a simple algorithm to compute one time step of a numerical scheme, in which u^k serves as initial condition for computing (4.88) within each four-pixel cell up to time τ . Since this scheme is based on composing analytic solutions for small image patches, we will call it a *locally analytic scheme*. Its structure is depicted in Figure 4.4.

Stability Analysis. The values of the analytical solution (4.88) at arbitrary times $t \geq 0$ are convex combinations of its initial values. By its construction from the analytical solution (4.88), the novel scheme in Figure 4.4 therefore obeys the maximum-minimum principle. Consequently, it is absolutely stable for each τ .

Consistency Analysis. For simplicity, we analyse consistency in the case of TV flow, i.e. $p = 1$. While we have set the spatial step size to $h = 1$ throughout the

derivation of the scheme, we have to admit variable spatial step size h here. Then, by (4.89) our final scheme reads

$$\begin{aligned} \frac{u_{i,j}^{k+1} - u_{i,j}^k}{\tau} = & \frac{g_{i-1/2,j-1/2}}{h^2} \cdot (\mu_{i,j,-}^k - u_{i,j}^k) \min \left\{ 1, \frac{h^2}{4\tau g_{i-1/2,j-1/2}} \right\} \\ & + \frac{g_{i+1/2,j-1/2}}{h^2} \cdot (\mu_{i,j,+-}^k - u_{i,j}^k) \min \left\{ 1, \frac{h^2}{4\tau g_{i+1/2,j-1/2}} \right\} \\ & + \frac{g_{i-1/2,j+1/2}}{h^2} \cdot (\mu_{i,j,-+}^k - u_{i,j}^k) \min \left\{ 1, \frac{h^2}{4\tau g_{i-1/2,j+1/2}} \right\} \\ & + \frac{g_{i+1/2,j+1/2}}{h^2} \cdot (\mu_{i,j,++}^k - u_{i,j}^k) \min \left\{ 1, \frac{h^2}{4\tau g_{i+1/2,j+1/2}} \right\}. \end{aligned} \quad (4.94)$$

This scheme can be considered as a stabilisation of the explicit scheme (4.93). It coincides with (4.93), and is therefore a consistent approximation of order $O(\tau + h^2)$ for TV diffusion, if each of the four minimum operations on its right-hand side selects the value 1. This consistency condition is fulfilled for

$$0 \leq \tau \leq \min \left\{ \frac{h^2}{4g_{i-1/2,j-1/2}}, \frac{h^2}{4g_{i+1/2,j-1/2}}, \frac{h^2}{4g_{i-1/2,j+1/2}}, \frac{h^2}{4g_{i+1/2,j+1/2}} \right\}. \quad (4.95)$$

For larger τ it is again easy to see that linear diffusion $\partial_t u = \Delta u$ is approximated. However, this happens in regions where the gradient is already close to zero. In this case, the visual differences between linear diffusion and TV diffusion are small.

Boundary conditions. Neumann boundary conditions, also characterised as reflecting or zero-flux boundary conditions, which are the most common choice in PDE-based image filtering, can easily be implemented in the LAS framework. As in many discretisations of total variation flow and other isotropic diffusion filters, it is sufficient in the case of a rectangular $N \times M$ image domain to mirror before each time step the first and last rows and columns of the image and apply the same time step as for inner pixels also for boundary pixels with the so prepared neighbourhood.

For instance, in the four-pixel cell $\{0, 1\} \times \{j, j + 1\}$ where $1 \leq j, j + 1 \leq M$ we have due to the mirroring $u_{0,j}^k = u_{1,j}^k$ and $u_{0,j+1}^k = u_{1,j+1}^k$. Thus, one easily checks that the evolution (4.88) of the four-pixel cell conserves the equalities $u_{0,j} = u_{1,j}$ and $u_{0,j+1} = u_{1,j+1}$ for all times, and in particular mimicks a two-pixel dynamics on $u_{1,j}$ and $u_{1,j+1}$ that equals (4.14) for TV flow, or an analogous equation for $p \neq 1$. As a consequence, no transport (flux) across the image boundary separating columns 0 and 1 takes place, warranting zero-flux boundary conditions as desired.

Similarly, at an image corner, e.g. $\{0, 1\} \times \{0, 1\}$, the cell arising from the mirroring contains four equal pixels, leading to trivial dynamics that does not cause any boundary-crossing transport neither.



Figure 4.5: *Top left: (a)* Original image, 93×93 pixels. *Top right: (b)* TV diffusion with standard explicit scheme, where TV diffusivity is regularised with $\varepsilon = 0.01$, $\tau = 0.0025$, 10000 iterations. *Bottom left: (c)* TV diffusion computed with LAS without regularisation of diffusivity, $\tau = 0.0025$, 10000 iterations. *Bottom right: (d)* LAS with $\tau = 0.1$, 250 iterations.

Experiments. We illustrate our LAS by three experiments. First, in Figure 4.5, we contrast the regularisation-free LAS for TV diffusion with a standard explicit discretisation. In the latter scheme, TV diffusivity is approximated by the regularised TV diffusivity $1/\sqrt{|\nabla u|^2 + \varepsilon^2}$. Since the stability condition for explicit schemes imposes the upper limit $\varepsilon/4$ on the time step size, a high number of iterations is needed for a reasonably small ε . It can be seen that the LAS based on four-pixel discretisations and the unregularised TV diffusivity – which cannot be used in the explicit scheme – considerably reduce blurring effects caused by the discretisation.

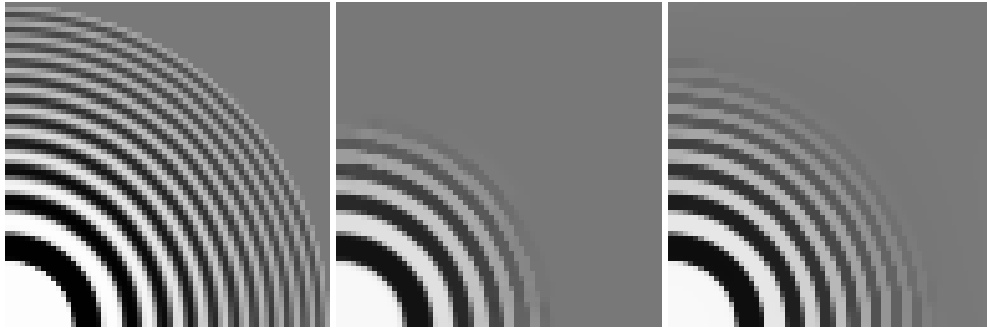


Figure 4.6: *Left: (a)* One quadrant of a rotationally invariant test image, 64×64 pixels. *Middle: (b)* TV diffusion with standard explicit scheme, $\varepsilon = 0.01$, $\tau = 0.0025$, 12000 iterations. *Right: (c)* TV diffusion with our LAS algorithm, $\tau = 0.1$, 300 iterations.



Figure 4.7: *Left:* Original image, 93×93 pixels. *Middle:* Balanced forward-backward diffusion with standard explicit scheme, $\varepsilon = 0.1$, $\tau = 0.0025$, 160000 iterations. *Right:* BFB diffusion with our LAS, $\tau = 0.1$, 4000 iterations.

Figure 4.6 demonstrates the reasonable rotational invariance of our LAS, which is achieved in spite of the fact that the analytically solvable case of the 2×2 -pixel cell is not the one with optimal rotational invariance.

Figure 4.7 demonstrates balanced forward-backward diffusion. With equal parameters, it can be seen again that the LAS looks sharper by preserving finer details. Moreover, it is worth mentioning that in the LAS experiments we were able to use a time step size that exceeded the largest admissible step size of the explicit scheme by a factor of 40.

4.2.6 Semi-Analytic Solution for Arbitrary Diffusivities

For general diffusivity functions $g(|\nabla u_\sigma|^2)$, as well as for the singular diffusivity functions (2.26) with discretisation parameters $\alpha \neq 1/2$, we do in general not have an analytic solution in the sense of (4.88) which incorporates the dynamics of g . Nevertheless, it is still possible to solve analytically the dynamical system (4.67), which governs the diffusion process within one four-pixel cell, for a *fixed* diffusivity field g .

To this end it is useful to introduce new variables $w_{i,j}$ by

$$W := HUH, \quad (4.96)$$

where $U := \begin{pmatrix} u_{1,1} & u_{2,1} \\ u_{1,2} & u_{2,2} \end{pmatrix}$, $W := \begin{pmatrix} w_{1,1} & w_{2,1} \\ w_{1,2} & w_{2,2} \end{pmatrix}$, and H happens to be the same matrix as introduced by (4.63). In terms of the new variables, (4.94) can be rewritten as

$$\begin{aligned} \dot{w}_{1,1} &= 0, & \dot{w}_{2,1} &= -4g w_{2,1}, \\ \dot{w}_{1,2} &= -4g w_{1,2}, & \dot{w}_{2,2} &= -8\alpha g w_{2,2}, \end{aligned} \quad (4.97)$$

where we have set $g := g_{3/2,3/2}$ for the only diffusivity value involved. The solution of (4.97) is easy to state: While $w_{1,1}$ stays constant, the other variables are subject to independent exponential decays,

$$\begin{aligned} w_{1,1}(t) &= w_{1,1}(0), & w_{2,1}(t) &= e^{-4gt} w_{2,1}(0), \\ w_{1,2}(t) &= e^{-4gt} w_{1,2}(0), & w_{2,2}(t) &= e^{-8\alpha gt} w_{2,2}(0). \end{aligned} \quad (4.98)$$

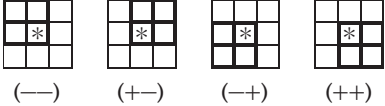
Via the inverse transform of (4.96),

$$U(t) = H W(t) H, \quad (4.99)$$

this solution can be rewritten in terms of the original variables.

LSAS Algorithm for Isotropic Diffusion

- Compute the pre-smoothed image $v := K_\sigma * u^k$ by convolution.
- For each four-pixel cell $\{i, i+1\} \times \{j, j+1\}$, compute the squared gradient $|\nabla v|^2$ according to (4.72), and the diffusivity $g = g(|\nabla v|^2)$.
- For each four-pixel cell, compute one time step of isotropic diffusion via the analytical solution (4.96), (4.98), (4.99).
- For each pixel (*) with coordinates (i, j) , consider the four cells



(---) (+-) (-+) (++)

which lead to four approximations

$$u_{i,j,---}^{k+1}, u_{i,j,+-}^{k+1}, u_{i,j,-+}^{k+1}, u_{i,j,++}^{k+1}.$$

Average:

$$u_{i,j}^{k+1} = \frac{1}{4}(u_{i,j,---}^{k+1} + u_{i,j,+-}^{k+1} + u_{i,j,-+}^{k+1} + u_{i,j,++}^{k+1}).$$

Figure 4.8: One time step of the locally semi-analytic scheme for a nonlinear isotropic diffusion process.

4.2.7 Locally Semi-Analytic Scheme for Isotropic Nonlinear Diffusion

Modifying the idea of our locally analytic scheme, we can use the solution given by (4.96), (4.98), (4.99) as an approximation to the isolated evolution in a four-pixel cell. Consequently, four solutions of this kind can again be used to approximate the solution of (4.65). In contrast to the previous case, it is now necessary to compute anew in each time step the pre-smoothed image $v = u_\sigma$ and the diffusivity field $g(|\nabla v|^2)$. One time step of the so adapted scheme is given in Figure 4.8. Note that this scheme can be used for arbitrary diffusivity functions g and for every discretisation parameter $\alpha \in [0, 1]$.

In analogy to semi-implicit schemes that keep the diffusivity fixed at the previous time level while discretising the remainder in an implicit fashion, we will call this scheme a *locally semi-analytic scheme (LSAS)*.

Stability. By substituting (4.99) into (4.98) one checks that, as in the LAS case, the values of the analytical solution for u for any $t \geq 0$ are convex combinations of the initial values. Via the averaging procedure this guarantees a maximum–minimum principle, and absolute stability of our LSAS for arbitrary time step size τ .

Consistency. Like the LAS, the locally semi-analytic scheme is conditionally consistent. For simplicity, we restrict our analysis to the case of the 1D nonlinear isotropic diffusion equation $\partial_t u = \partial_x(g \partial_x u)$ with bounded diffusivity function $g = g(|\partial_x v|^2)$. The LSAS with variable spatial step size h and time step size τ can then be written as

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \frac{1}{h} \left(\delta_{i+1/2}^k \frac{u_{i+1}^k - u_i^k}{h} - \delta_{i-1/2}^k \frac{u_i^k - u_{i-1}^k}{h} \right) \quad (4.100)$$

with

$$\delta_{i\pm 1/2}^k := \frac{h^2}{4\tau} \left(1 - \exp\left(-\frac{4\tau}{h^2} g_{i\pm 1/2}^k\right) \right). \quad (4.101)$$

Using the Taylor series of the exponential function, we see that

$$\delta_{i\pm 1/2}^k = g_{i\pm 1/2}^k + \mathcal{O}\left(\frac{\tau}{h^2}\right) \quad (4.102)$$

from which it follows that the LSAS is an approximation of the underlying PDE of order $\mathcal{O}(\tau + h^2 + \frac{\tau}{h^2})$. Therefore, it is consistent if $\frac{\tau}{h^2} \rightarrow 0$ as $\tau, h \rightarrow 0$. The latter consistency condition constitutes a difference to the LAS situation. Nevertheless, it is uncritical in the application context of digital image processing that we have in view: Here, the grid size is in fact fixed to $h = 1$, and the LSAS is an unconditionally consistent approximation to the space-discrete dynamical system.

Let us also characterise the error that occurs at large time step sizes. It is easy to see that for $\tau \rightarrow \infty$, the scheme creates constant 2×2 patches, except in case $\alpha = 0$ where

the local averaging is restricted to the components of a checkerboard decomposition. Averaging the patches approximates linear diffusion with a constant diffusivity.

Boundary conditions. Reflecting Neumann boundary conditions for a rectangular image domain are implemented simply by mirroring the first and last rows and columns of the image. The arguments demonstrating the zero-flux property of the so obtained process are the same as in the case of the LAS for singular isotropic diffusion processes. In this respect our LSAS behaves much like any other explicit scheme working on a 3×3 stencil around each pixel.

4.2.8 Multi-Channel Images

The generalisation of nonlinear isotropic diffusion to multi-channel images has been described in 2.2.5: The divergence expression $\operatorname{div}(g \cdot \nabla u)$ is considered for each image channel separately but using one and the same common diffusivity g that depends on all channels.

This procedure can be used straightforward with the discretisations developed in this section. To extend the discretisation of the squared gradient (4.72) to a multi-channel image $v = (v_{k;i,j})_{i,j,k}$ where $v_{k;i,j}$ is the value of the k -th image channel at pixel (i, j) , we use

$$G_\alpha(v) := \sum_k G_\alpha(v_k) \quad (4.103)$$

with $G_\alpha(v_k)$ given by (4.72) as squared gradient approximation in the argument of g . One uses then for each channel separately the divergence discretisation (4.65) or any of the derived schemes, inserting the single diffusivity field g in all channels.

In the cases in which the isotropic nonlinear diffusion process can be represented as gradient descent of an energy, this is also true for the discretised multi-channel process. A suitable discrete energy function is obtained by inserting (4.103) in the place of the squared gradient in (4.73).

4.2.9 Haar Wavelet Interpretation

Like its 1D counterpart, also our locally analytic scheme for 2D singular isotropic diffusion, as well as the locally semi-analytic schemes for general 2D isotropic diffusion can be described in terms of single-scale Haar wavelet shrinkage procedures. The underlying observation is that the finest scale Haar wavelet transform acts in a natural way on 2×2 -pixel cells.

Haar wavelet shrinkage on a four-pixel image. We describe the action of Haar wavelet shrinkage on one single four-pixel tile $F := (f_{i,j})_{i,j=1}^2$. One cycle of Haar wavelet shrinkage consists of three steps.

1. *Analysis step.* The four-pixel image F is transformed into the wavelet domain by applying low and high pass Haar filters to its rows and columns. More precisely, F is multiplied from the left and the right by the matrix H from (4.63) which results in

$$C = \begin{pmatrix} c_{1,1} & c_{2,1} \\ c_{1,2} & c_{2,2} \end{pmatrix} := HFH. \quad (4.104)$$

Setting $U = F$ and $W = C$, this is exactly the variable transform (4.96).

2. *Shrinkage step.* The high-pass coefficients of C are modified by reducing the absolute values of some or all of them. To this end, we apply a shrinkage function S_θ depending on a threshold parameter θ to the high-pass filtered coefficients, i.e., we compute $S_\theta(c_{1,2}), S_\theta(c_{2,1}), S_\theta(c_{2,2})$, while leaving the low-pass coefficient $c_{1,1}$ fixed.
3. *Synthesis step.* The inverse transform of step 1 is used to take the shrunken coefficients back from the wavelet to the signal domain,

$$F^{(1)} = HS_\theta(C)H. \quad (4.105)$$

This is just the analog of (4.99).

Shrinkage functions. In conventional wavelet shrinkage, thresholding depends on the individual coefficients. For example, *soft shrinkage* [75] shrinks the coefficients towards 0 by an amount that is given by a threshold parameter θ :

$$S_\theta(c_{i,j}) := \begin{cases} c_{i,j} - \theta \operatorname{sgn}(c_{i,j}) & \text{if } |c_{i,j}| \geq \theta, \\ 0 & \text{otherwise.} \end{cases} \quad (4.106)$$

In [144] a shrinkage function inspired by isotropic nonlinear diffusion filtering was introduced that leads to a *coupled* shrinking of the coefficients. More precisely, the thresholding applies with respect to $\gamma(C) := (c_{2,1}^2 + c_{1,2}^2 + c_{2,2}^2)^{1/2}$. For a soft shrinkage and $(i, j) \in \{(2, 1), (1, 2), (2, 2)\}$ this comes down to

$$S_\theta(c_{i,j}) := \begin{cases} c_{i,j} - \frac{\theta}{\gamma(C)} \operatorname{sgn}(c_{i,j}) & \text{if } \gamma(C) \geq \theta, \\ 0 & \text{otherwise.} \end{cases} \quad (4.107)$$

In contrast to the classical wavelet shrinkage, this results in an improved rotation invariance of the resulting image.

We follow this idea and define our shrinkage function in dependence on

$$G(F) = (c_{1,2}^2 + c_{2,1}^2 + c_{2,2}^2)^{1/2}. \quad (4.108)$$

It is straightforward to check that the value $G(F)$ indeed coincides with $G_{1/2}(F)$ as defined in (4.72).

Shrinkage function for singular diffusivities. Applying the shrinkage function

$$S_\theta(s; G(F)) := \begin{cases} (1 - 4p (G(F))^{-p} \theta)^{1/p} s, & G(F) \geq (4p \theta)^{1/p}, \\ 0 & G(F) < (4p \theta)^{1/p}. \end{cases} \quad (4.109)$$

our Haar wavelet shrinkage produces for $i, j = 1, 2$ the values

$$f_{i,j}^{(1)} = \begin{cases} \mu + (1 - 4p (G(F))^{-p} \theta)^{1/p} (f_{i,j} - \mu), & G(F) \geq (4p \theta)^{1/p}, \\ \mu, & G(F) < (4p \theta)^{1/p}. \end{cases} \quad (4.110)$$

Comparing this equation with (4.88) we observe that on 2×2 pixels our Haar wavelet shrinkage with shrinkage function (4.109) coincides with the solution of the nonlinear diffusion equation with diffusivity (2.26), where the shrinkage parameter θ plays the same role as the diffusion time t .

Shrinkage function for arbitrary diffusivities. Moreover, by directly translating (4.98) into the shrinkage function

$$\begin{aligned} S_\theta(c_{2,1}) &:= e^{-4g\theta} c_{2,2}, \\ S_\theta(c_{1,2}) &:= e^{-4g\theta} c_{1,2}, \\ S_\theta(c_{2,2}) &:= e^{-8g\alpha\theta} c_{2,2} \end{aligned} \quad (4.111)$$

and using this instead of (4.109), we obtain another shrinkage process which is equivalent to our semi-analytic solution from Subsection 4.2.7. Note that in this case a coupling between all three wavelet coefficients is mediated by the common quantity g . The coupling is restricted to the first two wavelet coefficients only for $\alpha = 0$, when $c_{2,2}$ is left unshrunk.

The discretisations of the diffusion process and diffusivities leading to our LAS and LSAS are in full analogy to those used in [144, 145] to derive coupled Haar wavelet shrinkage processes with improved rotational invariance. The shrinkage rules in [144] equal the first-order terms of (4.111).

Single-scale Haar wavelet shrinkage on the entire image. Given an $N \times M$ -pixel image f , ordinary single scale Haar wavelet shrinkage divides the image into disjoint 2×2 -pixel cells and performs Haar wavelet shrinkage on each of these cells as

described above. Unfortunately, this process is neither shift invariant nor rotation invariant. However, both properties can be achieved with a little more effort by the following *2D cycle-spinning* procedure:

1. Shift the original image $f_{++} := (f_{i,j})$ one pixel to the right to obtain $f_{-+} := (f_{i-1,j})$, one pixel down to get $f_{+-} := (f_{i,j-1})$ and one pixel to the right and down resulting in $f_{--} := (f_{i-1,j-1})$,
2. Perform wavelet shrinkage (4.110) on the 2×2 cells of the four images $f_{--}, f_{+-}, f_{-+}, f_{++}$, i.e., four times ordinary Haar wavelet shrinkage.
3. Shift the resulting images back and compute the average.

It is obvious that the 2D cycle spinning mimicks exactly the way how the analytic or semi-analytic solutions of the dynamical systems (4.67) are combined into solutions of (4.65). Consequently, it describes exactly one time step of size $\tau = \theta$ of our locally analytic scheme (4.94) if the shrinkage function (4.109) is used, or of our locally semi-analytic scheme if the shrinkage function (4.111) is chosen.

4.3 Four-Pixel Approach to Anisotropic Nonlinear Diffusion

In this section, we are concerned with the nonlinear anisotropic diffusion equation (2.35) with anisotropic diffusion tensor $D(J)$ depending on the structure tensor (2.36). To reduce numerical blurring effects as much as possible, we derive a discretisation of (2.35) which can again be split into approximations on the four-pixel cells (4.66). We will then also discretise (2.36) within such cells.

4.3.1 Discretisation of the Divergence Form

Let us first discretise the anisotropic diffusion equation

$$\partial_t u = \operatorname{div}(D \cdot \nabla u) \quad (4.112)$$

with an arbitrary diffusion tensor field represented by positive semidefinite symmetric matrices $D = \begin{pmatrix} a & c \\ c & b \end{pmatrix}$. We assume that u is sampled at the integer pixel positions (i, j) while D is sampled at inter-pixel positions $(i + 1/2, j + 1/2)$.

First approximation. To discretise the right-hand side of (4.112) at a given pixel position (i, j) , we will use values of u at positions $(i + \varepsilon_1, j + \varepsilon_2)$ with $\varepsilon_1, \varepsilon_2 \in \{-1, 0, +1\}$, together with the diffusion tensors at $(i \pm 1/2, j \pm 1/2)$. We use the abbreviations

$$\begin{aligned} D_{i-1/2, j-1/2} &:= \begin{pmatrix} a_{--} & c_{--} \\ c_{--} & b_{--} \end{pmatrix}, & D_{i+1/2, j-1/2} &:= \begin{pmatrix} a_{+-} & c_{+-} \\ c_{+-} & b_{+-} \end{pmatrix}, \\ D_{i-1/2, j+1/2} &:= \begin{pmatrix} a_{-+} & c_{-+} \\ c_{-+} & b_{-+} \end{pmatrix}, & D_{i+1/2, j+1/2} &:= \begin{pmatrix} a_{++} & c_{++} \\ c_{++} & b_{++} \end{pmatrix}. \end{aligned} \quad (4.113)$$

Expanding the differential operators div and ∇ into partial derivatives ∂_x, ∂_y , we calculate

$$\begin{aligned} \left(\operatorname{div}(D \nabla u) \right)_{i,j} &= \left((\partial_x, \partial_y) D (\partial_x, \partial_y)^T u \right)_{i,j} \\ &= \left(\partial_x (a \partial_x u + c \partial_y u) + \partial_y (c \partial_x u + b \partial_y u) \right)_{i,j} \\ &\approx (a \partial_x u + c \partial_y u)_{i+1/2, j} - (a \partial_x u + c \partial_y u)_{i-1/2, j} \\ &\quad + (c \partial_x u + b \partial_y u)_{i, j+1/2} - (c \partial_x u + b \partial_y u)_{i, j-1/2}. \end{aligned} \quad (4.114)$$

Interpolating entries of D and image values linearly between the locations where they are given, we continue

$$\begin{aligned}
& \left(\operatorname{div}(D \nabla u) \right)_{i,j} \\
& \approx \frac{a_{++} + a_{+-}}{2} (u_{i+1,j} - u_{i,j}) + \frac{(c \partial_y u)_{i+1/2,j+1/2} + (c \partial_y u)_{i+1/2,j-1/2}}{2} \\
& - \frac{a_{-+} + a_{--}}{2} (u_{i,j} - u_{i-1,j}) - \frac{(c \partial_y u)_{i-1/2,j+1/2} + (c \partial_y u)_{i-1/2,j-1/2}}{2} \\
& + \frac{(c \partial_x u)_{i+1/2,j+1/2} + (c \partial_x u)_{i-1/2,j+1/2}}{2} + \frac{b_{++} + b_{-+}}{2} (u_{i,j+1} - u_{i,j}) \\
& - \frac{(c \partial_x u)_{i+1/2,j-1/2} + (c \partial_x u)_{i-1/2,j-1/2}}{2} - \frac{b_{+-} + b_{--}}{2} (u_{i,j} - u_{i,j-1}).
\end{aligned} \tag{4.115}$$

Approximating $(\partial_y u)_{i+1/2,j+1/2}$ by $\frac{1}{2}(u_{i+1,j+1} - u_{i+1,j} + u_{i,j+1} - u_{i,j})$ etc., we are led to the dynamical system

$$\begin{aligned}
\dot{u}_{i,j} = & \frac{1}{2}(-a_{--}(u_{i,j} - u_{i-1,j}) - b_{--}(u_{i,j} - u_{i,j-1}) - c_{--}(u_{i,j} - u_{i-1,j-1})), \\
& + \frac{1}{2}(a_{+-}(u_{i+1,j} - u_{i,j}) - b_{+-}(u_{i,j} - u_{i,j-1}) - c_{+-}(u_{i+1,j-1} - u_{i,j})), \\
& + \frac{1}{2}(-a_{-+}(u_{i,j} - u_{i-1,j}) + b_{-+}(u_{i,j+1} - u_{i,j}) + c_{-+}(u_{i,j} - u_{i-1,j+1})), \\
& + \frac{1}{2}(a_{++}(u_{i+1,j} - u_{i,j}) + b_{++}(u_{i,j+1} - u_{i,j}) + c_{++}(u_{i+1,j+1} - u_{i,j})),
\end{aligned} \tag{4.116}$$

where we have denoted differentiation with respect to t by a dot.

We observe again that each summand on the right-hand side combines only quantities from one of the four-pixel cells (4.66), see Figure 4.3, giving way to splitting up (4.116) into the average of four dynamical systems each of which is localised in a single cell.

Denoting the diffusion tensor discretised in $(3/2, 3/2)$ simply by $D = \begin{pmatrix} a & c \\ c & b \end{pmatrix}$, we state such a four-pixel dynamical system for the cell $\{1, 2\} \times \{1, 2\}$:

$$\begin{aligned}
\dot{u}_{1,1} &= 2a(u_{2,1} - u_{1,1}) + 2b(u_{1,2} - u_{1,1}) + 2c(u_{2,2} - u_{1,1}), \\
\dot{u}_{2,1} &= 2a(u_{1,1} - u_{2,1}) + 2b(u_{2,2} - u_{2,1}) - 2c(u_{1,2} - u_{2,1}), \\
\dot{u}_{1,2} &= 2a(u_{2,2} - u_{1,2}) + 2b(u_{1,1} - u_{1,2}) - 2c(u_{2,1} - u_{1,2}), \\
\dot{u}_{2,2} &= 2a(u_{1,2} - u_{2,2}) + 2b(u_{2,1} - u_{2,2}) + 2c(u_{1,1} - u_{2,2}).
\end{aligned} \tag{4.117}$$

Second approximation. As in Subsection 4.2.1, we can use representations of div and ∇ in terms of the 45° -rotated coordinates ξ, η for an alternative discretisation. The diffusion tensor D then needs to be transformed into

$$HDH^T = \frac{1}{2} \begin{pmatrix} a+b+2c & a-b \\ a-b & a+b-2c \end{pmatrix}. \tag{4.118}$$

This leads to

$$\begin{aligned}
\left(\operatorname{div}(D \nabla u)\right)_{i,j} &= \left((\partial_\xi, \partial_\eta)(HDH(\partial_\xi u, \partial_\eta u)^T)\right)_{i,j} \\
&= \frac{1}{2} \left(\partial_\xi((a+b+2c)\partial_\xi u + (a-b)\partial_\eta u) \right. \\
&\quad \left. + \partial_\eta((a-b)\partial_\xi u + (a+b-2c)\partial_\eta u) \right)_{i,j} \\
&\approx \frac{1}{2\sqrt{2}} \left(((a+b+2c)\partial_\xi u + (a-b)\partial_\eta u)_{i+1/2,j+1/2} \right. \\
&\quad - ((a+b+2c)\partial_\xi u + (a-b)\partial_\eta u)_{i-1/2,j-1/2} \\
&\quad + ((a-b)\partial_\xi u + (a+b-2c)\partial_\eta u)_{i+1/2,j-1/2} \\
&\quad \left. - ((a-b)\partial_\xi u + (a+b-2c)\partial_\eta u)_{i-1/2,j+1/2} \right). \tag{4.119}
\end{aligned}$$

Substituting central differences for $\partial_\xi u$ and $\partial_\eta u$ yields the dynamical system

$$\begin{aligned}
\dot{u}_{i,j} &= \frac{1}{4} \left(-(a_{--} + b_{--} + 2c_{--})(u_{i,j} - u_{i-1,j-1}) \right. \\
&\quad - (a_{--} - b_{--})(u_{i,j-1} - u_{i-1,j}) \\
&\quad + (a_{+-} - b_{+-})(u_{i+1,j} - u_{i,j-1}) \\
&\quad + (a_{+-} + b_{+-} - 2c_{+-})(u_{i+1,j-1} - u_{i,j}) \\
&\quad - (a_{-+} - b_{-+})(u_{i,j+1} - u_{i-1,j}) \\
&\quad - (a_{-+} + b_{-+} - 2c_{-+})(u_{i,j} - u_{i-1,j+1}) \\
&\quad + (a_{++} + b_{++} + 2c_{++})(u_{i+1,j+1} - u_{i,j}) \\
&\quad \left. + (a_{++} - b_{++})(u_{i+1,j} - u_{i,j+1}) \right). \tag{4.120}
\end{aligned}$$

Like (4.116), this system is the average of four-pixel systems which for the cell $\{1,2\} \times \{1,2\}$ read in this case

$$\begin{aligned}
\dot{u}_{1,1} &= (a+b+2c)(u_{2,2} - u_{1,1}) + (a-b)(u_{2,1} - u_{1,2}), \\
\dot{u}_{2,1} &= (a+b-2c)(u_{1,2} - u_{2,1}) + (a-b)(u_{1,1} - u_{2,2}), \\
\dot{u}_{1,2} &= (a+b-2c)(u_{2,1} - u_{1,2}) + (a-b)(u_{2,2} - u_{1,1}), \\
\dot{u}_{2,2} &= (a+b+2c)(u_{1,1} - u_{2,2}) + (a-b)(u_{1,2} - u_{2,1}). \tag{4.121}
\end{aligned}$$

As in the isotropic case, any convex combination

$$\begin{aligned}
\dot{u}_{1,1} &= q_\alpha(u_{2,1} - u_{1,1}) + r_\alpha(u_{1,2} - u_{1,1}) + s_\alpha(u_{2,2} - u_{1,1}), \\
\dot{u}_{2,1} &= q_\alpha(u_{1,1} - u_{2,1}) + r_\alpha(u_{2,2} - u_{2,1}) - s_\alpha(u_{1,2} - u_{2,1}), \\
\dot{u}_{1,2} &= q_\alpha(u_{2,2} - u_{1,2}) + r_\alpha(u_{1,1} - u_{1,2}) - s_\alpha(u_{2,1} - u_{1,2}), \\
\dot{u}_{2,2} &= q_\alpha(u_{1,2} - u_{2,2}) + r_\alpha(u_{2,1} - u_{2,2}) + s_\alpha(u_{1,1} - u_{2,2}), \tag{4.122}
\end{aligned}$$

of (4.117) and (4.121), where the coefficients q_α , r_α and s_α are given by

$$\begin{aligned} q_\alpha &:= (1 + \alpha)a + (1 - \alpha)b, \\ r_\alpha &:= (1 - \alpha)a + (1 + \alpha)b, \\ s_\alpha &:= (1 - \alpha)(a + b) + 2c, \end{aligned} \tag{4.123}$$

is also a consistent discretisation of anisotropic diffusion.

Inserting the *isotropic* diffusion tensor $D = g_{3/2,3/2}I$, where I is the 2×2 unit matrix, into (4.122) yields exactly the system (4.67) which underlines that our discretisation of anisotropic diffusion is a natural extension of the isotropic case.

4.3.2 Discretisation of the Diffusion Tensor

To discretise the diffusion tensor D , we need to discretise the structure tensor J . The most direct approach to do this is to discretise the gradients ∇v of the given pre-smoothed image $v := K_\sigma * u$. Much as in the case of isotropic diffusion, Section 4.2, the centre of a four-pixel cell is an outstanding location for discretising these quantities most locally. Let us therefore discretise $\nabla v = (\partial_x v, \partial_y v)^T$, and then $\nabla v \nabla v^T$, at the centre $(3/2, 3/2)$ of a four-pixel cell $\{v_{ij}\}_{i,j=1,2}$ from a sampling of the spatial function v .

First approximation. Considering four pixels $v_{11}, v_{12}, v_{21}, v_{22}$ belonging to a quadratic grid with mesh size 1, we approximate $\partial_x v$ and $\partial_y v$ at the midpoints between neighbouring pixel positions and take arithmetic means of these expressions to obtain approximations for the derivatives at $(3/2, 3/2)$:

$$\begin{aligned} (\partial_x v)_{3/2,3/2} &\approx \frac{1}{2}(v_{2,2} + v_{2,1} - v_{1,2} - v_{1,1}), \\ (\partial_y v)_{3/2,3/2} &\approx \frac{1}{2}(v_{2,2} - v_{2,1} + v_{1,2} - v_{1,1}). \end{aligned} \tag{4.124}$$

Taking the outer product $\nabla v \nabla v^T$ from the so discretised gradient ∇v leads to the four components

$$\begin{aligned} (\partial_x v)_{3/2,3/2}^2 &\approx \frac{1}{4}(v_{2,2} + v_{2,1} - v_{1,2} - v_{1,1})^2 \\ (\partial_y v)_{3/2,3/2}^2 &\approx \frac{1}{4}(v_{2,2} - v_{2,1} + v_{1,2} - v_{1,1})^2 \\ (\partial_x v \partial_y v)_{3/2,3/2} &\approx \frac{1}{4}(v_{2,2} + v_{2,1} - v_{1,2} - v_{1,1})(v_{2,2} - v_{2,1} + v_{1,2} - v_{1,1}) \\ &= \frac{1}{4}((v_{2,2} - v_{1,1})^2 - (v_{2,1} - v_{1,2})^2). \end{aligned} \tag{4.125}$$

Componentwise Gaussian convolution with K_ϵ yields the approximated structure tensor field.

Second approximation. An alternative way to discretise J uses one-sided finite differences at each of the four pixels of the cell. At $(1, 1)$, one obtains the approximations

$$\begin{aligned} (\partial_x v)_{1,1} &\approx v_{2,1} - v_{1,1}, \\ (\partial_y v)_{1,1} &\approx v_{1,2} - v_{1,1}. \end{aligned} \quad (4.126)$$

From these an approximation of the outer product $\nabla v \nabla v^T$ at $(1, 1)$ is calculated. At the other three pixels, analogous approximations are made using left- or right-sided differences such as to stay within the given four-pixel cell. Taking the arithmetic mean of the four outer product approximations yields the approximation

$$\begin{aligned} (\partial_x v)_{3/2,3/2}^2 &\approx \frac{1}{2}((v_{2,2} - v_{1,2})^2 + (v_{2,1} - v_{1,1})^2) \\ (\partial_y v)_{3/2,3/2}^2 &\approx \frac{1}{2}((v_{2,2} - v_{2,1})^2 + (v_{1,2} - v_{1,1})^2) \\ (\partial_x v \partial_y v)_{3/2,3/2} &\approx \frac{1}{4}((v_{2,2} - v_{1,1})^2 - (v_{2,1} - v_{1,2})^2). \end{aligned} \quad (4.127)$$

Note that the two approximations (4.125) and (4.127) differ only in the main diagonal entries.

Once more, one can also use convex combinations of (4.125) and (4.127), i.e.,

$$\begin{aligned} (\partial_x v)_{3/2,3/2}^2 &\approx \frac{1}{4}((v_{2,2} + v_{2,1} - v_{1,2} - v_{1,1})^2 + \alpha(v_{2,2} - v_{2,1} - v_{1,2} + v_{1,1})^2) \\ (\partial_y v)_{3/2,3/2}^2 &\approx \frac{1}{4}((v_{2,2} - v_{2,1} + v_{1,2} - v_{1,1})^2 + \alpha(v_{2,2} - v_{2,1} - v_{1,2} + v_{1,1})^2) \\ (\partial_x v \partial_y v)_{3/2,3/2} &\approx \frac{1}{4}((v_{2,2} - v_{1,1})^2 - (v_{2,1} - v_{1,2})^2) \end{aligned} \quad (4.128)$$

for $\alpha \in [0, 1]$. An easy calculation shows that the trace $(\partial_x v)_{3/2,3/2}^2 + (\partial_y v)_{3/2,3/2}^2$ of the so discretised structure tensor equals the discretised isotropic diffusivity (4.72) for $i = j = 1$. The one-sided difference procedure used to obtain (4.127) could equally be used in the isotropic case to derive (4.70) in a different way.

4.3.3 Multi-Channel Case

Continuing the principle stated in Subsection 4.2.8, in such a process the channel evolutions are synchronised by sharing the diffusion tensor (which is simply the sum of the diffusion tensors of the separate channels) while the divergence expression acts on each channel separately.

It is worth noting that unlike for scalar-valued images for which (4.112) with $\varrho = 0$ and $\sigma = 0$ degenerates into an isotropic process, these parameters still admit a truly anisotropic process for multi-channel images.

4.3.4 Variational Interpretation in the Multi-Channel Case

In the multi-channel case with $\varrho = 0$, $\sigma = 0$, it is also possible to describe anisotropic diffusion as gradient descent for the variational problem of minimising a certain en-

ergy functional [214].³ In this subsection, we show that the discretisations (4.116), (4.120) and their weighted averages for the divergence expression arise naturally from gradient descents for energy functions in the mentioned special case and are put into a one-to-one correspondence to the approximations (4.125), (4.127) and their weighted averages for the gradient outer product. Note that in this situation also the structure tensor and the outer product (summed over channels) coincide.

We consider energy functions given by

$$E(u) := \frac{1}{2} \sum_{i,j} \text{tr} \Psi \left(\sum_k [\nabla u_k \nabla u_k^T]_{i+1/2, j+1/2} \right) \quad (4.129)$$

where the inner sum runs over the image channels u_k . As in Subsection 4.2.8, we will use $u_{k;i,j}$ to refer to the value of pixel (i, j) in u_k . Further, $[\nabla u_k \nabla u_k^T]_{i+1/2, j+1/2}$ is some discretisation of the gradient outer product $\nabla u_k \nabla u_k^T$ within the four-pixel cell $\{i, i+1\} \times \{j, j+1\}$, and the outer sum runs over all such four-pixel cells. By Ψ we denote an increasing function on \mathbb{R}_0^+ which is applied to its symmetric positive semidefinite matrix argument as usual by acting on the eigenvalues and leaving the eigenvector system intact.

According to the chain rule, the partial derivative of the energy E with respect to some particular variable $u_{k;i,j}$ is given by

$$\frac{\partial E}{\partial u_{k;i,j}} = \frac{1}{2} \sum_{i',j'} \text{tr} \left(\Psi' \left(\sum_l [\nabla u_l \nabla u_l^T]_{i'+1/2, j'+1/2} \right) \frac{\partial [\nabla u_k \nabla u_k^T]_{i'+1/2, j'+1/2}}{\partial u_{k;i,j}} \right), \quad (4.130)$$

the corresponding gradient descent by

$$\dot{u}_{k;i,j} = - \frac{\partial E}{\partial u_{k;i,j}} \quad \text{for all } i, j, k. \quad (4.131)$$

Similar as in the isotropic case, Subsection 4.2.3, Ψ' appears in the gradient descent, this time in the role of the function that ties the diffusion tensor D to the structure tensor J . For given (i, j) , the sum on the right-hand side has again four possibly nonzero contributions belonging to the cells (4.66), compare Figure 4.3. When specifying a discretisation $[\nabla u_l \nabla u_l^T]_{i'+1/2, j'+1/2}$ in (4.129), the same discretisation reappears in the argument of Ψ' in (4.131), but it also determines the expression for $\partial [\nabla u_k \nabla u_k^T]_{i'+1/2, j'+1/2} / \partial u_{k;i,j}$. We study the two discretisations from Subsection 4.3.2. In equations which do not contain explicit interactions between image channels, we will abbreviate $v := u_k$.

³For $\varrho \neq 0$ and $\sigma \neq 0$, no energy functional is known that warrants (4.112) as gradient descent.

First, by discretising the outer product analog to (4.125), we find that

$$\begin{aligned}
\frac{\partial[\nabla v \nabla v^T]_{(--)}}{\partial v_{i,j}} &= \frac{v_{i,j} - v_{i-1,j-1}}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{v_{i,j-1} - v_{i-1,j}}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \\
\frac{\partial[\nabla v \nabla v^T]_{{+})}}{\partial v_{i,j}} &= \frac{v_{i,j} - v_{i+1,j-1}}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{v_{i,j-1} - v_{i+1,j}}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \\
\frac{\partial[\nabla v \nabla v^T]_{{--)}}{\partial v_{i,j}} &= \frac{v_{i,j} - v_{i-1,j+1}}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{v_{i,j+1} - v_{i-1,j}}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \\
\frac{\partial[\nabla v \nabla v^T]_{{++)}}{\partial v_{i,j}} &= \frac{v_{i,j} - v_{i+1,j+1}}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{v_{i,j+1} - v_{i+1,j}}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
\end{aligned} \tag{4.132}$$

where we have used again the cell abbreviations from (4.66). By plugging this into the right-hand side of (4.131) and evaluate the trace expression, we obtain exactly (4.120), except for the additional channel indices.

If we use instead (4.127) in discretising the structure tensor, (4.132) is replaced by

$$\begin{aligned}
\frac{\partial[\nabla v \nabla v^T]_{(--)}}{\partial v_{i,j}} &= \frac{1}{2} \begin{pmatrix} 2(v_{i,j} - v_{i-1,j}) & v_{i,j} - v_{i-1,j-1} \\ v_{i,j} - v_{i-1,j-1} & 2(v_{i,j} - v_{i,j-1}) \end{pmatrix}, \\
\frac{\partial[\nabla v \nabla v^T]_{{+})}}{\partial v_{i,j}} &= \frac{1}{2} \begin{pmatrix} 2(v_{i,j} - v_{i+1,j}) & -v_{i,j} + v_{i+1,j-1} \\ -v_{i,j} + v_{i+1,j-1} & 2(v_{i,j} - v_{i,j-1}) \end{pmatrix}, \\
\frac{\partial[\nabla v \nabla v^T]_{{--)}}{\partial v_{i,j}} &= \frac{1}{2} \begin{pmatrix} 2(v_{i,j} - v_{i-1,j}) & -v_{i,j} + v_{i-1,j+1} \\ -v_{i,j} + v_{i-1,j+1} & 2(v_{i,j} - v_{i,j+1}) \end{pmatrix}, \\
\frac{\partial[\nabla v \nabla v^T]_{{++)}}{\partial v_{i,j}} &= \frac{1}{2} \begin{pmatrix} 2(v_{i,j} - v_{i+1,j}) & v_{i,j} - v_{i+1,j+1} \\ v_{i,j} - v_{i+1,j+1} & 2(v_{i,j} - v_{i,j+1}) \end{pmatrix}
\end{aligned} \tag{4.133}$$

which implies (4.116) as gradient descent.

To summarise, the derivation from the energy minimisation framework, though exactly valid only in the multi-channel image case with $\varrho = 0$, $\sigma = 0$, clearly indicates that the discretisations (4.120) and (4.125) correspond to each other, as do (4.116) and (4.127). This correspondence obviously extends to the weighted averages given by (4.122) and (4.128) with equal α .

4.3.5 Semi-Analytic Solution

Under the assumption that the diffusion tensors D are kept fixed during the image evolution, both systems (4.117) and (4.121) have fairly simple analytic solutions which can be exploited to establish numerical schemes. We want now to derive these solutions.

We use again the variable transform (4.96) to rewrite the weighted average (4.122) as

$$\begin{aligned}\dot{w}_{1,1} &= 0, & \dot{w}_{2,1} &= -4aw_{2,1} - 4cw_{1,2}, \\ \dot{w}_{1,2} &= -4cw_{2,1} - 4bw_{1,2}, & \dot{w}_{2,2} &= -4\alpha(a+b)w_{2,2}.\end{aligned}\quad (4.134)$$

This system decomposes into independent evolutions for $w_{1,1}$, which is constant, for $w_{2,2}$ which follows a one-dimensional linear dynamical system with the solution

$$w_{2,2}(t) = e^{-4\alpha(a+b)t} w_{2,2}(0) \quad (4.135)$$

(in particular, for $\alpha = 0$ it is constant) and the two-dimensional system

$$\dot{w} = -4Dw. \quad (4.136)$$

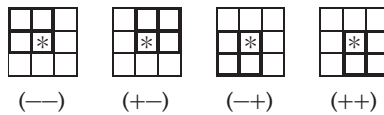
for $w := (w_{2,1}, w_{1,2})^T$. Let the eigendecomposition of D be given by $D = \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T$ with eigenvalues $\lambda_{1,2} = \frac{1}{2}(a+b \pm \sqrt{(a-b)^2 + 4c^2})$ and orthonormal eigenvectors e_1, e_2 . Then, remembering that D is kept constant, the solution of (4.136) is

$$w(t) = e^{-4\lambda_1 t} (e_1^T w(0)) e_1 + e^{-4\lambda_2 t} (e_2^T w(0)) e_2. \quad (4.137)$$

Via the inverse transform (4.99) the analytic solution given by (4.135), (4.137) can easily be expressed with respect to the original variables.

LSAS Algorithm for Anisotropic Diffusion

- Compute the pre-smoothed image $v := K_\sigma * u^k$ by convolution.
- For each four-pixel cell $\{i, i+1\} \times \{j, j+1\}$, compute the approximation of the tensor product $\nabla v \nabla v^T$ according to (4.128).
- Compute the structure tensor field $J = K_\varrho * (\nabla v \nabla v^T)$ by convolution.
- For each four-pixel cell, compute the diffusion tensor $D = D(J)$.
- For each four-pixel cell, compute one time step of anisotropic diffusion via the analytical solution (4.96), (4.135), (4.137), (4.99).
- For each pixel (*) with coordinates (i, j) , consider the four cells



which lead to four approximations

$$u_{i,j,--}^{k+1}, u_{i,j,+-}^{k+1}, u_{i,j,-+}^{k+1}, u_{i,j,++}^{k+1}.$$

Average:

$$u_{i,j}^{k+1} = \frac{1}{4}(u_{i,j,--}^{k+1} + u_{i,j,+-}^{k+1} + u_{i,j,-+}^{k+1} + u_{i,j,++}^{k+1}).$$

Figure 4.9: One time step of the locally semi-analytic scheme for anisotropic diffusion.

4.3.6 Locally Semi-Analytic Schemes for Anisotropic Nonlinear Diffusion

In an analogous manner as done in Subsection 4.2.7 we can employ the explicit solutions (4.135), (4.137) to construct numerical schemes for anisotropic diffusion. We use the splitting of (4.116) or (4.120) into the contributions from four-pixel cells. For these, the analytic solution derived in Subsection 4.3.5 is evaluated with evolution time equalling the time step size τ .

For the anisotropic diffusion processes in question, the diffusion tensor D , whose entries are needed to evaluate (4.135) and (4.137), depends on the structure tensor J_{ϱ} . In each time step, the evaluation of the dynamical systems has therefore to be preceded by recomputing D and its eigendecomposition from the current data u . This completes our LSAS, one step of which is summarised in Figure 4.9.

Stability. As for the LSAS for isotropic diffusion, Subsection 4.2.7, the use of our analytical solutions ensures that the resulting schemes for our four-pixel cell are stable in the Euclidean norm for any time step size (note that $\lambda_1, \lambda_2 \geq 0$). The averaging of four-pixel cells guarantees the transfer of the absolute stability to the entire grid.

Consistency. The consistency analysis sketched for the isotropic case before can be transferred to the anisotropic situation, showing that the LSAS, too, is a conditionally consistent approximation of order $O(\tau + h^2 + \frac{\tau}{h^2})$ to the corresponding PDE. It is therefore consistent if $\tau/h^2 \rightarrow 0$ as $\tau, h \rightarrow 0$. For a fixed spatial grid size h , as usual in image processing, it is an unconditionally consistent $O(\tau)$ approximation of the space-discrete dynamical system.

Let us briefly discuss the error behaviour for large time step sizes. Assuming a fixed spatial grid size and a time step size tending to infinity, a scheme based on (4.121) (i.e., $\alpha = 0$) approaches a local averaging on a *checkerboard decomposition* of our grid. For $\alpha \neq 0$, the averaging involves all pixels of the grid, though for $\alpha < 1$ the averaging is faster within each checkerboard component than between them.

Boundary conditions. Like its isotropic counterpart before, our LSAS for anisotropic diffusion enables the realisation of Neumann boundary conditions in a very simple manner by mirroring the first and last rows and columns of a rectangular $N \times M$ image domain. The four-pixel dynamics is then equally applied in the proper four-pixel cells lying completely within the image domain as well as in those four-pixel cells overlapping the boundary.

While in the isotropic case this comes as no surprise, as other explicit schemes allow the same procedure, it constitutes a real simplification in the anisotropic situation,

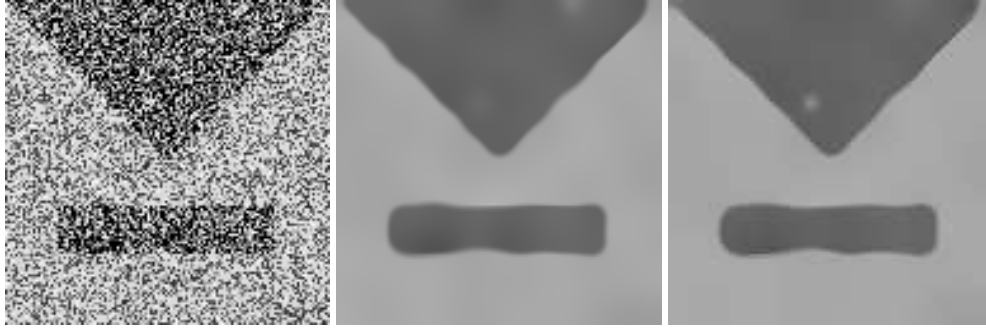


Figure 4.10: *Left to right:* (a) Test image with impulsive noise, 128×128 pixels. (b) Denoised by edge-enhancing diffusion, $\lambda = 5$, $\sigma = 1.8$, $\varrho = 0$, with standard explicit scheme, $\tau = 0.166$, $N = 200$ iterations. (c) Denoised by edge-enhancing diffusion, same parameters as in (b), but with LSAS, $\alpha = 0$, $\tau = 1$, $N = 200$ iterations.

since explicit discretisations that discretise the diffusion tensor at integer (pixel) locations typically require a special treatment of boundary rows and columns to rule out fluxes across the boundary.

Let us shortly prove this favourable behaviour of our anisotropic LSAS. We start by one additional reasoning needed to treat the pre-smoothed gradients in the computation of the diffusion tensor. Realising that the mirroring of boundary rows and columns is in fact part of a $(2N, 2M)$ -periodic continuation u^* of the $N \times M$ image u on the entire plane, it becomes evident that also the convolution u_σ^* of u^* with a Gaussian of standard deviation σ shares the periodicity and symmetry properties of u^* . In particular, the rows and columns of u_σ^* adjacent to the image domain equal mirrored copies of the boundary rows and columns. Therefore a simple extension of u_σ by mirroring the boundary rows and columns is compatible with the same mirroring operation on u itself.

Consequentially, the structure tensor computed e.g. in a boundary cell $\{0, 1\} \times \{j, j+1\}$ where $1 \leq j, j+1 \leq M$ will automatically have its eigenvector system aligned to the coordinate directions, with a zero eigenvalue assigned to the x direction. As the diffusion tensor inherits the eigenvector orientations of the structure tensor, it is a diagonal matrix. Inserting this diagonal matrix (i.e., with $c = 0$) into the discretised divergence form (4.116) or (4.120), one easily sees that $\dot{u}_{1,j} + \dot{u}_{1,j+1} = \dot{u}_{0,j} + \dot{u}_{0,j+1} = 0$, in both cases, such that no flux across the boundary takes place. In the case of a corner cell like $\{0, 1\} \times \{0, 1\}$ the dynamics again becomes trivial since all four pixels have equal values.

Experiments. In Figure 4.10 we use our scheme to perform edge-enhancing diffusion [206], see 2.3.1.1. Remember that in this case there is no integration over the outer products, so $\varrho = 0$. The diffusion tensor D is given by $g(J) = g(\nabla v \nabla v^T)$

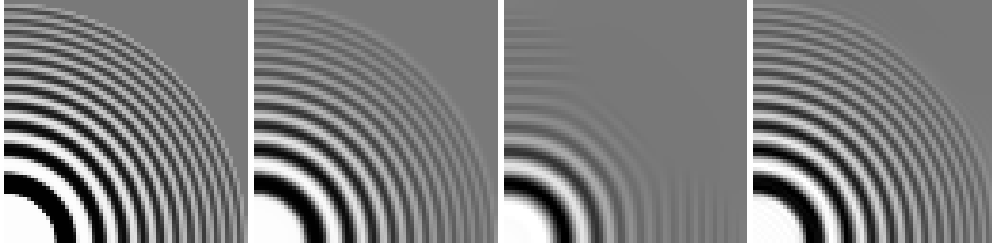


Figure 4.11: *Left to right:* (a) Rotationally invariant test image from Figure 4.6, 64×64 pixels. (b) Exact solution for coherence-enhancing diffusion, $\varepsilon = 0.001$, $C = 1$, $\sigma = 0.5$, $\varrho = 4$, at time $t = 250$. (c) Filtered with the nonnegativity scheme [206] with $\tau = 1/6$, and $N = 1500$ iterations. Average absolute error: 17.99. (d) Processed with our LSAS algorithm ($\alpha = 0$), same parameters. Average absolute error: 3.81.

where we use the function $g(s^2) = 1 - \exp(-3.31488\lambda^8/s^8)$ with a given threshold parameter $\lambda > 0$.

The noisy image, Figure 4.10(a), is denoised with a standard explicit scheme with central spatial differences, and with the locally semi-analytic scheme. It is observed that the denoising result with our new scheme is slightly sharper. Moreover, a look at the parameters shows that the effective evolution time used by the new scheme is six times larger than with the explicit scheme which demonstrates how much the latter is indeed dominated by numerical blurring artifacts.

In our next experiment, Figure 4.11, we consider coherence-enhancing diffusion (CED) [207], see 2.3.1.2, to perform smoothing along flow-like structures. For our rotationally invariant test image only radial linear diffusion with diffusivity ε takes place. Hence, the exact solution at time t is given by a convolution with a Gaussian of standard deviation $\sqrt{2\varepsilon t}$. By comparing the solutions of the so-called nonnegativity discretisation from [206] with our LSAS algorithm and the exact solution, we see that the LSAS scheme does not suffer from visible blurring artifacts. It preserves rotation invariance very well and creates significantly lower errors than the nonnegativity scheme.

These quantitative findings are also confirmed in the fingerprint example in Figure 4.12. We observe that the LSAS scheme gives much sharper results, and that it yields still realistic results for time step sizes far beyond the stability limit $1/6$ of the nonnegativity scheme.

Finally, we investigate the effect of different discretisation parameters α in our LSAS for anisotropic diffusion. The comparison of the CED filter results in Figure 4.13(a) and (b) reveals that, unlike in the isotropic case, the choice $\alpha = 0$ is clearly superior to $\alpha = 1$ in terms of both sharpness and rotational invariance. However, from Figure 4.13(c) and (d) it can be seen that introducing a small amount of the second



Figure 4.12: *Left to right:* (a) Fingerprint image, 100×100 pixels. (b) Filtered with CED, $\varepsilon = 0.001$, $C = 1$, $\sigma = 0.5$, $\varrho = 4$, using the nonnegativity scheme [206] with $\tau = 1/6$ and $N = 60$ iterations. (c) Processed with our LSAS scheme ($\alpha = 0$) for CED, same parameters. (d) LSAS scheme with $\tau = 1$ and $N = 10$ iterations.

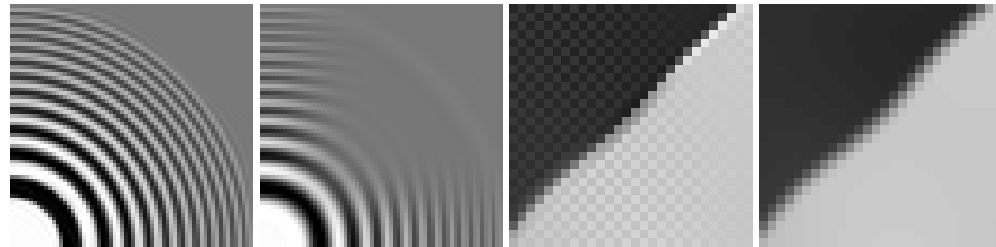


Figure 4.13: *Left to right:* (a) Test image from Figure 4.6(a) filtered by coherence-enhancing diffusion, $\varepsilon = 0.001$, $C = 1$, $\sigma = 0.5$, $\varrho = 4$, with LSAS, $\tau = 0.166$ and $N = 300$. The discretisation weight in the LSAS is $\alpha = 0$. (b) Same but with discretisation weight $\alpha = 1$. (c) Detail (28×28 pixels) from Figure 4.10(c), where the grey-values have been rescaled to raise contrast. (d) Corresponding detail from LSAS filtering result for CED with same parameters as in Figure 4.10(c) except $\alpha = 0.01$.

discretisation (1...2 percent) allows to suppress the checkerboard artifacts that are inherent to the $\alpha = 0$ method.

4.3.7 Haar Wavelet Interpretation

In analogy to Subsection 4.2.9 we want to introduce an *anisotropic shrinkage* procedure with respect to a diffusion tensor D . To do this, we only need to specify a suitable shrinkage function $S_\theta(C)$. In accordance with (4.135) and (4.137), our shrinkage function couples the reduction of the antidiagonal coefficients $c_{1,2}$ and $c_{2,1}$ while shrinking $c_{2,2}$ independently:

$$\begin{aligned}
 S_\theta \left(\begin{pmatrix} c_{2,1} \\ c_{1,2} \end{pmatrix} \right) &:= Q \begin{pmatrix} e^{-4\lambda_1\theta} & 0 \\ 0 & e^{-4\lambda_2\theta} \end{pmatrix} Q^T \begin{pmatrix} c_{2,1} \\ c_{1,2} \end{pmatrix}, \\
 S_\theta(c_{2,2}) &:= e^{-4\alpha(a+b)\theta} c_{2,2},
 \end{aligned} \tag{4.138}$$

where $Q := (e_1, e_2)$ denotes the eigenvector matrix of D , and the threshold parameter θ was identified with the diffusion time t . Let us abbreviate this anisotropic shrinkage procedure by $S_\theta(C)$.

With this shrinkage function, one cycle of the above anisotropic Haar wavelet shrinkage on a four-pixel patch F coincides with the solution of (4.122) with initial condition $U(0) = F$, where the threshold parameter plays the role of the diffusion time. By the same procedure as in the isotropic case, this extends into an equivalence of the LSAS algorithm and a shift-invariant single scale Haar wavelet shrinkage process.

Apart from shift invariance, the LSAS algorithm can also be regarded as a simple approach to create *rotationally invariant* anisotropic Haar wavelet shrinkage: Since our novel anisotropic shrinkage rules are a numerical scheme for a rotationally invariant continuous diffusion filter, rotation invariance is approximated at no additional expense.

4.3.8 Variational Interpretation with Fixed Diffusivity

The semi-analytic solution considered in 4.3.5 motivates another way to look at the dynamical systems (4.117) and (4.121). Assume the diffusion tensor field D is prescribed at the locations $(i + 1/2, j + 1/2)$, and consider the energy function

$$E(u) := \frac{1}{2} \sum_{i,j} \text{tr} \left(D_{i+1/2, j+1/2} \cdot \sum_k [\nabla u_k \nabla u_k^T]_{i+1/2, j+1/2} \right). \quad (4.139)$$

Of course, this is exactly (4.129) with a linear penalty function $\Psi(A) = D \cdot A$. Computations analog to those in 4.3.5 confirm that (4.117) and (4.121) describe gradient descent for E provided that the discretisations (4.125) and (4.127) are used for the structure tensor, respectively. Within our terminology, the so described diffusion process should correctly be called *anisotropic (inhomogeneous) linear diffusion*.

The same interpretation can be used to cover as well the case of a diffusion tensor field that depends on the structure tensor. This means that the anisotropic nonlinear diffusion process is approximated in each instant by an anisotropic linear diffusion process. However, this means that the process is considered as a gradient descent for a different energy function in each instant because the diffusion tensor field changes with the data. Thus, it cannot be used to derive a theoretical result about uniqueness of minimisers. Nevertheless, this model is also capable of describing anisotropic diffusion with pre-smoothing in the diffusion tensor argument, i.e., $\varrho \neq 0$ and/or $\sigma \neq 0$, and thereby also truly anisotropic diffusion of single-channel images.

Moreover, this connection between anisotropic nonlinear diffusion and anisotropic inhomogeneous linear diffusion can be exploited to simplify the derivation of numerical schemes. We will use this approach in the next section.

A caveat that needs to be mentioned at this point refers to boundary conditions. Our analysis in 4.3.6 made substantial use of the way how the diffusion tensor in anisotropic diffusion depends on the image data. Thus, the simple compatibility between a locally semi-analytic scheme and Neumann boundary conditions transfers to the corresponding discretisations of anisotropic inhomogeneous linear diffusion only if the prescribed diffusion tensor fields obey the same restriction as the dependent diffusion tensor fields in 4.3.6: Diffusion tensors D in the centre points of boundary cells must have their eigenvector systems aligned with the boundary direction.

4.4 Higher-Dimensional Extensions

Our ideas described in the previous sections can be extended straightforward to the case of higher-dimensional image domains. We demonstrate just two important examples in three dimensions: first, the discretisation of isotropic nonlinear diffusion with $\alpha = 1/2$ which includes the important case of the LAS for total variation flow; second, the discretisation of anisotropic diffusion leading to a LSAS.

4.4.1 Locally Analytic Scheme for Total Variation Diffusion in 3D

The natural extension of our ideas to three-dimensional images involves dynamical systems for eight-voxel ($2 \times 2 \times 2$) bricks. Neighbours of a voxel within such a brick now come in three types: they can be adjacent via an edge, a face diagonal or a volume diagonal, adding degrees of freedom to the discretisation procedure. For simplicity, we restrict ourselves to generalising the important case $\alpha = 1/2$ which underlies our LAS from Subsection 4.2.5. In this case, we have $G_{1/2}(v)$ which is just the quarter sum of squared differences of all pixel pairs in the four-pixel cell, and the dynamical system (4.79). It is easy to check that in the case of an eight-voxel cell $(u_{i,j,l})_{i,j,l=1,2}$ the expression

$$G(v) := \frac{1}{16} \sum_{(i,j,l) \neq (i',j',l')} (v_{i,j,l} - v_{i',j',l'})^2 \quad (4.140)$$

is a consistent approximation for $(|\nabla v|^2)_{3/2,3/2,3/2}$, while

$$\dot{u}_{i,j,l} = 4g \cdot (\mu - u_{i,j,l}), \quad i, j, l = 1, 2, \quad (4.141)$$

with μ denoting the arithmetic mean of the eight initial values, consistently approximates $u_t = \operatorname{div}(g \cdot \nabla u)$. Derivation from these approximations of a LSAS or, with the diffusivity (2.26), even a LAS for 3-D isotropic nonlinear diffusion based on eight-voxel bricks is straightforward.

4.4.2 Anisotropic Diffusion in 3D

Even in the anisotropic case, our framework can easily be extended to the higher-dimensional situation. In the following we will demonstrate shortly how a discretisation of the structure tensor and the divergence form of the diffusion process can be devised in the three-dimensional situation. Thereby, we obtain a numerical scheme for a generic 3D anisotropic diffusion process.

For simplicity, we restrict ourselves from the beginning to the consideration of scalar-valued images on the eight-voxel cell $\{1, 2\}^3$. We sketch two discretisations of the outer product matrix $\nabla v \nabla v^T$ at the location $(3/2, 3/2, 3/2)$ that resemble (4.125) and (4.127). The extension to a numerical scheme on the full regular 3D grid is clear.

First approximation of the outer product. In analogy to (4.124), we approximate $\partial_x v$, $\partial_y v$ and $\partial_z v$ at the midpoints between neighbouring voxel positions and take arithmetic means of these expressions to obtain approximations for the derivatives at $(3/2, 3/2, 3/2)$:

$$\begin{aligned} (\partial_x v)_{3/2, 3/2, 3/2} &\approx \frac{1}{4}(v_{2,2,2} + v_{2,2,1} + v_{2,1,2} + v_{2,1,1} - v_{1,2,2} - v_{1,2,1} - v_{1,1,2} - v_{1,1,1}) \\ (\partial_y v)_{3/2, 3/2, 3/2} &\approx \frac{1}{4}(v_{2,2,2} + v_{2,2,1} - v_{2,1,2} - v_{2,1,1} + v_{1,2,2} + v_{1,2,1} - v_{1,1,2} - v_{1,1,1}) \\ (\partial_z v)_{3/2, 3/2, 3/2} &\approx \frac{1}{4}(v_{2,2,2} - v_{2,2,1} + v_{2,1,2} - v_{2,1,1} + v_{1,2,2} - v_{1,2,1} + v_{1,1,2} - v_{1,1,1}). \end{aligned} \quad (4.142)$$

Using these expressions, one is led to the following approximation of the entries of the outer product matrix:

$$\begin{aligned} (\partial_x v)_{3/2, 3/2, 3/2}^2 &\approx \frac{1}{16}(v_{2,2,2} + v_{2,2,1} + v_{2,1,2} + v_{2,1,1} - v_{1,2,2} - v_{1,2,1} - v_{1,1,2} - v_{1,1,1})^2 \\ (\partial_y v)_{3/2, 3/2, 3/2}^2 &\approx \frac{1}{16}(v_{2,2,2} + v_{2,2,1} - v_{2,1,2} - v_{2,1,1} + v_{1,2,2} + v_{1,2,1} - v_{1,1,2} - v_{1,1,1})^2 \\ (\partial_z v)_{3/2, 3/2, 3/2}^2 &\approx \frac{1}{16}(v_{2,2,2} - v_{2,2,1} + v_{2,1,2} - v_{2,1,1} + v_{1,2,2} - v_{1,2,1} + v_{1,1,2} - v_{1,1,1})^2 \\ (\partial_x v \partial_y v)_{3/2, 3/2, 3/2} &\approx \frac{1}{16} \left((v_{2,2,2} + v_{2,2,1} - v_{1,1,2} - v_{1,1,1})^2 \right. \\ &\quad \left. - (v_{2,1,2} + v_{2,1,1} - v_{1,2,2} - v_{1,2,1})^2 \right) \\ (\partial_x v \partial_z v)_{3/2, 3/2, 3/2} &\approx \frac{1}{16} \left((v_{2,2,2} + v_{2,1,2} - v_{1,2,1} - v_{1,1,1})^2 \right. \\ &\quad \left. - (v_{2,2,1} + v_{2,1,1} - v_{1,2,2} - v_{1,1,2})^2 \right) \\ (\partial_y v \partial_z v)_{3/2, 3/2, 3/2} &\approx \frac{1}{16} \left((v_{2,2,2} - v_{2,1,1} + v_{1,2,2} - v_{1,1,1})^2 \right. \\ &\quad \left. - (v_{2,1,2} - v_{2,2,1} + v_{1,1,2} - v_{1,2,1})^2 \right). \end{aligned} \quad (4.143)$$

Second approximation of the outer product. Once more, we obtain a second approximation of $\nabla v \nabla v^T$ by starting from separate discretisations in all eight voxels $(1, 1, 1), \dots, (2, 2, 2)$ and averaging. In voxel $(1, 1, 1)$, the discretisation is based on the one-sided difference approximations

$$\begin{aligned} (\partial_x v)_{1,1,1} &\approx v_{2,1,1} - v_{1,1,1}, \\ (\partial_y v)_{1,1,1} &\approx v_{1,2,1} - v_{1,1,1}, \\ (\partial_z v)_{1,1,1} &\approx v_{1,1,2} - v_{1,1,1}. \end{aligned} \quad (4.144)$$

The partial derivatives are discretised in an analog manner in the remaining seven voxels, using left- or right-sided differences as appropriate to stay within $\{1, 2\}^3$. Computing outer products in all eight voxels and averaging yields

$$\begin{aligned} (\partial_x v)_{3/2,3/2,3/2}^2 &\approx \frac{1}{4} \left((v_{2,1,1} - v_{1,1,1})^2 + (v_{2,1,2} - v_{1,1,2})^2 \right. \\ &\quad \left. + (v_{2,2,1} - v_{1,2,1})^2 + (v_{2,2,2} - v_{1,2,2})^2 \right), \\ (\partial_y v)_{3/2,3/2,3/2}^2 &\approx \frac{1}{4} \left((v_{1,2,1} - v_{1,1,1})^2 + (v_{1,2,2} - v_{1,1,2})^2 \right. \\ &\quad \left. + (v_{2,2,1} - v_{2,1,1})^2 + (v_{2,2,2} - v_{2,1,2})^2 \right), \\ (\partial_z v)_{3/2,3/2,3/2}^2 &\approx \frac{1}{4} \left((v_{1,1,2} - v_{1,1,1})^2 + (v_{1,2,2} - v_{1,2,1})^2 \right. \\ &\quad \left. + (v_{2,1,2} - v_{2,1,1})^2 + (v_{2,2,2} - v_{2,2,1})^2 \right), \\ (\partial_x v \partial_y v)_{3/2,3/2,3/2} &\approx \frac{1}{8} \left((v_{2,2,1} - v_{1,1,1})^2 - (v_{2,1,1} - v_{1,2,1})^2 \right. \\ &\quad \left. + (v_{2,2,2} - v_{1,1,2})^2 - (v_{2,1,2} - v_{1,2,2})^2 \right), \\ (\partial_x v \partial_z v)_{3/2,3/2,3/2} &\approx \frac{1}{8} \left((v_{2,1,2} - v_{1,1,1})^2 - (v_{2,1,1} - v_{1,1,2})^2 \right. \\ &\quad \left. + (v_{2,2,2} - v_{1,2,1})^2 - (v_{2,2,1} - v_{1,2,2})^2 \right), \\ (\partial_y v \partial_z v)_{3/2,3/2,3/2} &\approx \frac{1}{8} \left((v_{1,2,2} - v_{1,1,1})^2 - (v_{1,2,1} - v_{1,1,2})^2 \right. \\ &\quad \left. + (v_{2,2,2} - v_{2,1,1})^2 - (v_{2,2,1} - v_{2,1,2})^2 \right). \end{aligned} \quad (4.145)$$

Discretisation of the divergence form. Each of the approximations (4.143), (4.145) and convex combinations of them can be used in calculating the structure tensor field of a given image. Moreover, each of them (or convex combinations) can be used to derive a compatible approximation of the divergence form of anisotropic diffusion.

To this end, we use the variational interpretation of an anisotropic diffusion process with prescribed diffusivity as discussed in 4.3.8 for the 2D case. Adapted to the 3D,

single-channel case and reduced to one eight-voxel cell, the energy reads

$$E(u) = \frac{1}{2} \text{tr}(D \cdot [\nabla u \nabla u^T]_{3/2,3/2,3/2}) \quad (4.146)$$

where

$$D = D_{3/2,3/2,3/2} = \begin{pmatrix} a & d & e \\ d & b & f \\ e & f & c \end{pmatrix} \quad (4.147)$$

is the single diffusion tensor value prescribed in the cell centre.

The gradient descent that approximates anisotropic diffusion within the eight-voxel cell in each time step of our numerical scheme is then given by

$$\dot{u}_{i,j,k} = -8 \frac{\partial}{\partial u_{i,j,k}} E(u) = -4 \text{tr} \left(D \cdot \frac{\partial}{\partial u_{i,j,k}} [\nabla u \nabla u^T]_{3/2,3/2,3/2} \right). \quad (4.148)$$

The scaling factor 8 (that does not change the property of being a gradient descent) reflects the periodic boundary condition due to which each voxel is treated as if being under the influence of eight identical (up to reflections) cells.

First approximation of the divergence form. Inserting (4.143) into (4.148) leads to the dynamical system

$$\dot{u} = F(D)u \quad (4.149)$$

with $u := (u_{1,1,1}, u_{1,1,2}, u_{1,2,1}, u_{1,2,2}, u_{2,1,1}, u_{2,1,2}, u_{2,2,1}, u_{2,2,2})^T$ and

$$F(D) = 1/2 \cdot \begin{pmatrix} -a-b-c & -a+b+c & -a+b-c & -a+b+c & a-b-c & a-b+c & a+b-c & a+b+c \\ -2d-2e-2f & -2d & -2e & +2f & -2f & +2e & +2d & +2d+2e+2f \\ -a-b+c & -a-b-c & -a+b+c & -a+b-c & a-b+c & a-b-c & a+b+c & a+b-c \\ -2d & -2d+2e+2f & -2f & +2e & -2e & +2f & +2d-2e-2f & +2d \\ -a+b-c & -a+b+c & -a-b-c & -a-b+c & a+b-c & a+b+c & a-b-c & a-b+c \\ -2e & -2f & +2d-2e+2f & +2d & -2d & -2d+2e-2f & +2f & +2e \\ -a+b+c & -a+b-c & -a-b+c & -a-b-c & a+b+c & a+b-c & a-b+c & a-b-c \\ +2f & +2e & +2d & +2d+2e-2f & -2d-2e+2f & -2d & -2e & -2f \\ a-b-c & a-b+c & a+b-c & a+b+c & -a-b-c & -a-b+c & -a+b-c & -a+b+c \\ -2f & -2e & -2d & -2d-2e+2f & +2d+2e-2f & +2d & +2e & +2f \\ a-b+c & a-b-c & a+b+c & a+b-c & -a-b+c & -a-b-c & -a+b+c & -a+b-c \\ +2e & +2f & -2d+2e-2f & -2d & +2d & +2d-2e+2f & -2f & -2e \\ a+b-c & a+b+c & a-b-c & a-b+c & -a+b-c & -a+b+c & -a-b-c & -a-b+c \\ +2d & +2d-2e-2f & +2f & -2e & +2e & -2f & -2d+2e+2f & -2d \\ a+b+c & a+b-c & a-b+c & a-b-c & -a+b+c & -a+b-c & -a-b+c & -a-b-c \\ +2d+2e+2f & +2d & +2e & -2f & +2f & -2e & -2d & -2d-2e-2f \end{pmatrix} \quad (4.150)$$

Second approximation of the divergence form. With the same notations, inserting (4.145) into (4.148) yields $\dot{u} = F(D)u$ with

$$F(D) = 1/2 \cdot \begin{pmatrix} \frac{-4(a+b+c)}{+d+e+f} & 4c & 4b & 4f & 4a & 4e & 4d & 0 \\ 4c & \frac{-4(a+b+c)}{+d-e-f} & -4f & 4b & -4e & 4a & 0 & 4d \\ 4b & -4f & \frac{-4(a+b+c)}{-d+e-f} & 4c & -4d & 0 & 4a & 4e \\ 4f & 4b & 4c & \frac{-4(a+b+c)}{-d-e+f} & 0 & -4d & -4e & 4a \\ 4a & -4e & -4d & 0 & \frac{-4(a+b+c)}{-d-e+f} & 4c & 4b & 4f \\ 4e & 4a & 0 & -4d & 4c & \frac{-4(a+b+c)}{-d+e-f} & -4f & 4b \\ 4d & 0 & 4a & -4e & 4b & -4f & \frac{-4(a+b+c)}{+d-e-f} & 4c \\ 0 & 4d & 4e & 4a & 4f & 4b & 4c & \frac{-4(a+b+c)}{+d+e+f} \end{pmatrix} \quad (4.151)$$

Remark. As in the 2D case, it is of course possible to use convex combinations of both approximations. It can be seen from the structure of (4.150) that using the pure first approximation again leads to a checkerboard-like decoupling which involves four interlaced components, each of which consists of voxels linked by spatial diagonals. In contrast, the second discretisation allows no direct transport along spatial diagonals at all.

Semi-analytic solutions. Much as in the 2D case a transform of the eight grey-values of the $2 \times 2 \times 2$ -cell into their wavelet coefficients simplifies the dynamical systems associated with the two discretisations. Setting

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{pmatrix} := \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix} u \quad (4.152)$$

the dynamical system (4.149), (4.150) transforms into

$$\begin{aligned} \dot{w}_0 &= 0 & \dot{w}_4 &= 0 \\ \begin{pmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{pmatrix} &= -4D \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} & \dot{w}_5 &= 0 \\ & & \dot{w}_6 &= 0 \\ & & \dot{w}_7 &= 0, \end{aligned} \quad (4.153)$$

while (4.149), (4.151) transforms into the decoupled systems

$$\begin{aligned} \dot{w}_0 &= 0 & \dot{w}_7 &= -4 \operatorname{tr}(D) w_7 \\ \begin{pmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{pmatrix} &= -4D \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} & \begin{pmatrix} \dot{w}_4 \\ \dot{w}_5 \\ \dot{w}_6 \end{pmatrix} &= -4\tilde{D} \begin{pmatrix} w_4 \\ w_5 \\ w_6 \end{pmatrix} \end{aligned} \quad (4.154)$$

with

$$\tilde{D} := \begin{pmatrix} a+b & f & e \\ f & a+c & d \\ e & d & b+c \end{pmatrix}. \quad (4.155)$$

For a convex combination consisting of $(1-\alpha)$ times the first and α times the second system one has therefore the analytic solution

$$\begin{aligned} w_0(t) &= w_0(0) & w_7(t) &= e^{-4\alpha t \operatorname{tr}(D)} w_7(0) \\ \begin{pmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{pmatrix} &= e^{-4tD} \begin{pmatrix} w_1(0) \\ w_2(0) \\ w_3(0) \end{pmatrix} & \begin{pmatrix} w_4(t) \\ w_5(t) \\ w_6(t) \end{pmatrix} &= e^{-4\alpha t \tilde{D}} \begin{pmatrix} w_4(0) \\ w_5(0) \\ w_6(0) \end{pmatrix} \end{aligned} \quad (4.156)$$

which can be computed using the eigendecompositions of D and \tilde{D} and gives rise to semi-analytic schemes for 3D anisotropic diffusion in analogy to those in 4.3.6.

Remark. The variety of possible discretisations of $\nabla v \nabla v^T$ within an eight-voxel cell of a 3D grid is much greater than in a four-pixel cell in 2D. The two possibilities shown here have been selected for their analogy to the 2D case. An evaluation of the suitability of these, and possibly other, numerical schemes for practical 3D image processing by meaningful experiments is beyond the scope of the present work.

4.5 Splitting Schemes

To end this chapter, we explain how the locally analytic and semi-analytic schemes fit into the framework of additive splitting schemes. We start by a short description of splitting schemes for the numerical solution of PDEs.

(This section has been revised in July 2013.)

4.5.1 Error Analysis

Consider an autonomous dynamical system

$$\dot{u} = B(u)u \quad (4.157)$$

where u is a spatially discretised image evolving in time t , the dot denotes the derivative w.r.t. t , and $B(u)$ is a matrix which depends on u . Systems of this type arise naturally as spatial discretisations of PDE-based image filters.

We assume for now that B is bounded,⁴

$$\|B(u)\| \leq M, \quad (4.158)$$

and satisfies a Lipschitz condition w.r.t. u ,

$$\|B(u_2) - B(u_1)\| \leq L \|u_2 - u_1\|. \quad (4.159)$$

However, the principles laid out here can often be transferred to cases with discontinuous right-hand sides, provided that the discontinuities are accounted for by an adequate concept of solution, as mentioned in Subsection 4.1.1.

For simplicity of notation we assume that u and all its approximations in the following are bounded by U , i.e. $\|u\| \leq U$.

If B is either constant, or u is one-dimensional, the exact evolution over the time interval $[t_1, t_2]$ is given by

$$u(t_2) = \exp\left(\int_{t_1}^{t_2} B(u(t)) dt\right) u(t_0). \quad (4.160)$$

Unfortunately, in all other cases – which means in particular, in all cases which are of real interest for us – the exact solution of (4.157) can not be stated in such an explicit form.

Nevertheless, (4.160) is a useful approximation for the exact solution of (4.157) for small time intervals. We will therefore refer to (4.160) as *exp-int approximation* in the following. The following proposition gives an error estimate for this approximation.

⁴Here and in the following, we use always the spectral norm $\|\cdot\| \equiv \|\cdot\|_2$ for matrices.

Proposition 4.4 Let u denote the solution of $\dot{u} = B(u(t))u$ on the interval $[t_1, t_2]$ of length τ , and \check{u} its continuous exp-int approximation according to

$$\check{u}(\tilde{t}) = \exp\left(\int_{t_1}^{\tilde{t}} B(\check{u}(t)) dt\right) u_0 \tag{4.161}$$

for $\tilde{t} \in [t_1, t_2]$ with the initial condition $\check{u}(t_1) = u(t_1) = u_0$. Then the error of the exp-int approximation is given by

$$e_0 = \|\check{u}(t_2) - u(t_2)\| = O(\tau^3). \tag{4.162}$$

The proof of this proposition will be provided in Subsection 4.5.4.

However, (4.160) is still an implicit description due to the occurrence of $u(t)$ in the integrand on the right hand side, and an explicit computation of this solution is generally impossible. Instead, one resorts to numerical time-stepping schemes that rely on different approximations of (4.160) for a short time interval of length $t_2 - t_1 = \tau$.

One possible simplification results if the matrix $B(u)$ admits a decomposition

$$B(u) = \frac{1}{R} \sum_{r=1}^R B_r(u) \tag{4.163}$$

into an average of operators B_r , whose actions, expressed by the dynamical systems

$$\dot{v} = B_r(v)v, \quad r = 1, \dots, R, \tag{4.164}$$

and their solutions are simpler to control than (4.160) itself. Numerical methods that capitalise on such decompositions are *splitting schemes* in a fairly general sense.

We mention here two important splittings of B . The *additive operator splitting* (AOS) scheme of [215], see also [134], uses a *directional* splitting where R equals the number of dimensions of the image domain (2 in a usual planar image), and the B_r correspond to 1-D diffusion processes in the basis directions. The schemes of this chapter, instead, rely on a decomposition of B into $R = 4$ (in the 2D case) matrices, each of which combines the four-pixel dynamics of the disjoint four-pixel cells of one of the four possible tilings of the grid with such cells. For instance, one of these matrices combines the dynamical systems of all cells $\{2i, 2i + 1\} \times \{2j, 2j + 1\}$. Each B_r can therefore be written (in a suitable ordering of rows and columns) as a block-diagonal matrix consisting of 4×4 blocks.

Using (4.164) one can rewrite (4.160) as

$$u(t_2) = \exp\left(\int_{t_1}^{t_2} \frac{1}{R} \sum_{r=1}^R B_r(u(t)) dt\right) u(t_1) = \exp\left(\frac{1}{R} \sum_{r=1}^R \int_{t_1}^{t_2} B_r(u(t)) dt\right) u(t_1) \tag{4.165}$$

which can be approximated using the Baker–Campbell–Hausdorff formula by

$$u(t_2) = \frac{1}{R} \sum_{r=1}^R \exp\left(\int_{t_1}^{t_2} B_r(u(t)) dt\right) u(t_1) + e_1 \quad (4.166)$$

with an error term $e_1 = O((t_2 - t_1)^2)$.⁵ The use of this additive decomposition of $\exp(\int B(u) dt)$ characterises *additive splitting methods*.

Still, each summand on the right hand side contains the evolving image $u(t)$ and thereby depends on the full evolution (4.160). A further approximation step replaces $B_r(u(t))$ by $B_r(u_{(r)}(t))$ where $u_{(r)}(t)$ denotes the modified image evolving by the dynamical system

$$\dot{u}_{(r)} = B_r(u_{(r)}) u_{(r)}, \quad t \in [t_1, t_2], \quad (4.167)$$

with the initial condition $u_{(r)}(t_1) := u(t_1)$. By Gronwall's Lemma, one proves that $\|u_{(r)}(t) - u(t)\| = O(t - t_1)$ for $t \in [t_1, t_2]$ and therefore

$$\begin{aligned} e_2 &:= \left\| \frac{1}{R} \sum_{r=1}^R \exp\left(\int_{t_1}^{t_2} B_r(u(t)) dt\right) u(t_1) - \frac{1}{R} \sum_{r=1}^R \exp\left(\int_{t_1}^{t_2} B_r(u_{(r)}(t)) dt\right) u(t_1) \right\| \\ &\leq \frac{1}{R} \sum_{r=1}^R \left\| \exp\left(\int_{t_1}^{t_2} B_r(u(t)) dt\right) - \exp\left(\int_{t_1}^{t_2} B_r(u_{(r)}(t)) dt\right) \right\| \cdot \|u(t_1)\| \\ &= \frac{1}{R} \sum_{r=1}^R \left\| \int_{t_1}^{t_2} (B_r(u(t)) - B_r(u_{(r)}(t))) dt + O((t_2 - t_1)^2) \right\| \cdot \|u(t_1)\| \\ &\leq \frac{1}{R} \|u(t_1)\| \cdot \sum_{r=1}^R \int_{t_1}^{t_2} \|B_r\| \cdot \|u(t) - u_{(r)}(t)\| dt + O((t_2 - t_1)^2) \\ &\leq \frac{1}{R} \|u(t_1)\| \cdot \sum_{r=1}^R (t_2 - t_1) \|B_r\| \cdot O(t_2 - t_1) + O((t_2 - t_1)^2) \\ &= O((t_2 - t_1)^2). \end{aligned} \quad (4.168)$$

Since both e_1 and e_2 are error contributions related to the splitting itself, they can be denoted as *splitting errors*. It should be noted that the exp-int approximation error e_0 is of third order while e_1 , e_2 and all other error contributions for numerical schemes discussed in the following are of second order, thus dominating e_0 . Acknowledging

⁵We remark that the contribution of this approximation to the error of a numerical scheme using the approximation (4.166) and time step size $\tau = t_2 - t_1$ is given by $e_1/\tau = O(\tau)$, since one has to consider the error of approximating $\partial_t u$ via $\frac{u(t_2) - u(t_1)}{t_2 - t_1}$. The same holds for all other error contributions discussed subsequently.

its presence in all of the subsequent approximations, we can therefore drop e_0 from the further discussions.

For a time-stepping scheme, one uses approximations of one time step of size τ ($= t_2 - t_1$) of (4.167), and averages them in each time step into a new approximation of $u(t_2)$ which then takes the role of $u(t_1)$ in the next time step. Additive splitting methods use different ways to approximate the numerical solutions of the systems (4.167).

Semi-implicit schemes. If in (4.167) the matrix $B_r(u_{(r)}(t))$ is approximated by $\tilde{B}_r := B_r(u(t_1))$, a linear dynamical system is obtained. An implicit Euler discretisation approximates $\exp(-(t_2 - t_1)\tilde{B}_r dt)$ in the solution

$$\exp(-(t_2 - t_1)\tilde{B}_r)u_{(r)}(t_2) = u(t_1) \quad (4.169)$$

by $I - (t_2 - t_1)\tilde{B}_r + O((t_2 - t_1)^2)$, thus leading to a system of linear equations,

$$(I - (t_2 - t_1)\tilde{B}_r)\tilde{u}_{(r)}(t_2) = u(t_1), \quad (4.170)$$

whose solution $\tilde{u}_{(r)}(t_2)$ approximates $u_{(r)}(t_2)$. Note that of the two splitting error contributions discussed above, only e_1 is encountered here, since e_2 is replaced by the (generally larger) error e_3 that results from keeping $B_{(r)}$ constant, and the linearisation error e_4 .

The semi-implicit approach underlies the (directional) additive operator splitting (AOS) scheme [134, 215] where the summands B_r correspond to discretised one-dimensional diffusion processes, as mentioned above. The resulting systems of equations are tridiagonal and admit a very efficient exact solution.

4.5.2 LAS and LSAS as Splitting Schemes

Semi-analytic schemes. Using the same constant approximation \tilde{B}_r for $B_r(u_{(r)}(t))$ one obtains

$$u_{(r)}(t_2) = \exp((t_2 - t_1)\tilde{B}_r)u(t_1). \quad (4.171)$$

If it is possible to compute the analytic solution for the right hand side of this system, a *semi-analytic scheme* results. Our LSAS algorithms in Subsections 4.2.7 and 4.3.6 are of this kind. Here we have $R = 4$ and the matrices B_r correspond to the four possible tilings of the image plane with disjoint four-pixel cells, thereby making these schemes *locally semi-analytic*.

Again, only e_1 occurs as splitting error, while e_2 is replaced by the same error e_3 mentioned for the semi-analytic scheme. No linearisation error e_4 is incurred.

Analytic schemes. If an analytic solution can be given even for the system (4.167), without additional approximation, we have an *analytic scheme*. Naturally, this is the most special case, and consequentially our LAS method in Subsection 4.2.5 had to be designed for a specific class of diffusivity functions.

Here, the splitting error comprises e_0 , e_1 and e_2 , while no errors e_3 and e_4 occur.

4.5.3 Locally Semi-Implicit Scheme for Isotropic Nonlinear Diffusion

Another way to exploit our local splitting would be to combine it with the procedure described above which leads to semi-implicit schemes. For example, in the case of isotropic nonlinear diffusion this would mean to linearise (4.67) or, simpler, (4.97) in time as

$$\begin{aligned} \frac{1}{\tau}(w_{1,1}^{k+1} - w_{1,1}^k) &= 0, & \frac{1}{\tau}(w_{2,1}^{k+1} - w_{2,1}^k) &= -4g^k w_{2,1}^{k+1}, \\ \frac{1}{\tau}(w_{1,2}^{k+1} - w_{1,2}^k) &= -4g^k w_{1,2}^{k+1}, & \frac{1}{\tau}(w_{2,2}^{k+1} - w_{2,2}^k) &= -8\alpha g^k w_{2,2}^{k+1}, \end{aligned} \quad (4.172)$$

where k and $k + 1$ refer to the old and new time step, resp., and τ is the time step size. Since the linear equations in the transformed variables $w_{i,j}$ decouple, one reads the solution direct from the equations:

$$\begin{aligned} w_{1,1}^{k+1} &= w_{1,1}^k, & w_{2,1}^{k+1} &= \frac{w_{2,1}^k}{1 + 4\tau g}, \\ w_{1,2}^{k+1} &= \frac{w_{1,2}^k}{1 + 4\tau g}, & w_{2,2}^{k+1} &= \frac{w_{2,2}^k}{1 + 8\alpha\tau g}. \end{aligned} \quad (4.173)$$

By using simply (4.173) in place of (4.98), our LSAS from Subsection 4.2.7 would be converted into a *locally semi-implicit scheme* for isotropic nonlinear diffusion.

4.5.4 Proof of the Error Bound for the Exp-Int Approximation

It remains to prove the error bound of Proposition 4.4. To this end, we will compare time-discrete versions of both (4.157) and (4.160) and prove their convergence to the time-continuous case.

For a more convenient presentation of these estimates, we introduce a change of notation at this point. The (time) interval of length τ that was denoted by $[t_1, t_2]$ in Sections 4.5.1–4.5.3, including Proposition 4.4 will from now on be referred to as $[t_0, t^*]$, in order to reserve t_1, t_2, \dots for the splitting points of subdivisions of this interval.

Convergence of time-discrete evolution to the time-continuous evolution. In a first step we prove that the error of an explicit time-discrete approximation of the exact solution of (4.157) on a given interval goes to zero with refined time steps.

Consider a subdivision of the time interval $[t_0, t^*]$ of length τ by equidistant points t_1, \dots, t_{N-1} according to $t_0 < t_1 < \dots < t_{N-1} < t_N = t^*$ with $t_{k+1} - t_k = \frac{\tau}{N}$. Let $N \geq 2$.

Let u be the exact solution of (4.157) with initial condition $u(t_0) = u_0$. By \tilde{u} we shall denote an approximate solution obtained by holding B constant within each subinterval $[t_k, t_{k+1}]$, i.e. the solution of

$$\dot{\tilde{u}} = \tilde{B}(t)\tilde{u} \quad (4.174)$$

with $\tilde{u}(t_0) = u_0$ and $\tilde{B}(t) = \tilde{B}_k = B(\tilde{u}(t_k))$ for $t_k \leq t < t_{k+1}$. It is known that

$$\tilde{u}_k = \exp\left(\frac{\tau}{N}\tilde{B}_{k-1}\right) \cdot \exp\left(\frac{\tau}{N}\tilde{B}_{k-2}\right) \cdot \dots \cdot \exp\left(\frac{\tau}{N}\tilde{B}_0\right)u_0. \quad (4.175)$$

Lemma 4.5 Assume that $\tau \leq \frac{N}{4L}$. Then the approximate solution \tilde{u} on $[t_0, t_N]$ satisfies

$$\|\tilde{u}(t) - u(t)\| \leq LMU \frac{\tau^2}{N} + \mathcal{O}\left(\frac{\tau^3}{N}\right) \quad (4.176)$$

for all $t \in [t_0, t_N]$.

Proof. By mathematical induction we prove that for $t \in [t_k, t_{k+1}]$ the inequality

$$\|\tilde{u}(t) - u(t)\| \leq kLMU^2 \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \quad (4.177)$$

holds.

For $k = 0$ we have $\tilde{u}(t_0) = u(t_0) = u_0$, thus (4.177) is fulfilled. For $k \geq 0$ let (4.177) hold up to k . Then we have for $t \in [t_k, t_{k+1}]$ that

$$\begin{aligned} \|\tilde{u}(t) - u(t)\| &\leq \|\tilde{u}(t) - \exp((t - t_k)B(\tilde{u}(t_k)))u(t_k)\| \\ &\quad + \|\exp((t - t_k)B(\tilde{u}(t_k)))u(t_k) - \exp((t - t_k)B(u(t_k)))u(t_k)\| \\ &\quad + \|\exp((t - t_k)B(u(t_k)))u(t_k) - u(t)\| \\ &= \|\exp((t - t_k)B(\tilde{u}(t_k)))(\tilde{u}(t_k) - u(t_k))\| \\ &\quad + \|(\exp((t - t_k)(B(\tilde{u}(t_k)) - B(u(t_k)))) - 1) \\ &\quad \quad \cdot \exp((t - t_k)B(u(t_k)))u(t_k)\| \\ &\quad + \|\exp((t - t_k)B(u(t_k)))u(t_k) - u(t)\| \end{aligned} \quad (4.178)$$

and by using the induction hypothesis in the first summand, induction hypothesis and Lipschitz condition in the second summand, and Gronwall's inequality in the third summand, we obtain

$$\begin{aligned}
\|\tilde{u}(t) - u(t)\| &\leq e^{\frac{M\tau}{N}} \left((k-1)LMU^2 \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \right) \\
&\quad + e^{\frac{\tau}{N} \cdot L \cdot (k-1)LMU^2 \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^4}{N^4}\right)} e^{\frac{M\tau}{N}} U \\
&\quad + \left(e^{\frac{\tau}{N} \cdot \frac{\tau}{N} LMU} - 1 \right) e^{\frac{M\tau}{N}} U \\
&= \left(1 + \mathcal{O}\left(\frac{\tau}{N}\right) \right) U \left(\left(\frac{(k-1)LMU}{N^2} \tau^2 + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \right) \right. \\
&\quad \left. + \left(\frac{(k-1)L^2MU^2}{N^3} \tau^3 + \mathcal{O}\left(\frac{k^2\tau^4}{N^4}\right) \right) \right. \\
&\quad \left. + \left(\frac{LMU^2}{N^2} \tau^2 + \mathcal{O}\left(\frac{\tau^4}{N^4}\right) \right) \right) \\
&= \frac{kLMU^2}{N^2} \tau^2 + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right), \tag{4.179}
\end{aligned}$$

where the higher order terms have been combined using

$$\mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) + \mathcal{O}\left(\frac{k\tau^3}{N^3}\right) + \mathcal{O}\left(\frac{k^2\tau^4}{N^4}\right) + \mathcal{O}\left(\frac{\tau^4}{N^4}\right) = \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right). \tag{4.180}$$

This ends the proof of the lemma. \square

The important consequence of Lemma 4.5 is that for a given interval $[t_0, t^*]$ of length τ the error of the discrete time-stepping solution \tilde{u} compared to the exact solution u vanishes with refinement of the subdivision, $N \rightarrow \infty$.

Error estimate for the discrete exp-int approximation. Our next step is to prove an error estimate for the discretisation of (4.160) with respect to the discretisation (4.174) of the original equation (4.157). The discrete time-stepping approximation of (4.160) with the same interval subdivision as before reads

$$\hat{u}(t_N) = \exp\left(\frac{\tau}{N} \sum_{j=0}^{N-1} B(\hat{u}(t_j))\right) u(t_0). \tag{4.181}$$

We introduce the following abbreviations:

$$\hat{B}_k := B(\hat{u}(t_k)), \quad (4.182)$$

$$\tilde{B}_k := B(\tilde{u}(t_k)), \quad (4.183)$$

$$E_k := \left\| \exp\left(\frac{\tau}{N} \sum_{j=0}^{k-1} \hat{B}_j\right) - \prod_{j=0}^{k-1} \exp\left(\frac{\tau}{N} \tilde{B}_j\right) \right\|, \quad (4.184)$$

$$Z_k := \exp\left(\frac{\tau}{N} \sum_{j=0}^{k-1} \hat{B}_j\right) - \prod_{j=0}^{k-1} \exp\left(\frac{\tau}{N} \hat{B}_j\right), \quad (4.185)$$

$$\Delta_k := \|Z_k\|, \quad (4.186)$$

$$D_k := \left\| \exp\left(\frac{\tau}{N} \hat{B}_k\right) - \exp\left(\frac{\tau}{N} \tilde{B}_k\right) \right\|. \quad (4.187)$$

Lemma 4.6 *One has*

$$E_N = \mathcal{O}(\tau^3). \quad (4.188)$$

Proof. Clearly, \hat{u} and \tilde{u} coincide on the first subinterval $[t_0, t_1]$, thus $\hat{B}_0 = \tilde{B}_0$, $\hat{B}_1 = \tilde{B}_1$ and therefore $D_0 = D_1 = 0$. Moreover, $\Delta_0 = 0$ and $E_0 = 0$. This is the basis for an inductive proof of

$$D_k = \mathcal{O}\left(\frac{k^3 \tau^4}{N^4}\right), \quad (4.189)$$

$$\Delta_k = \frac{\tau^3}{N^3} LM^2 U \binom{k+1}{3} + \mathcal{O}\left(\frac{k^3 \tau^4}{N^4}\right) \quad (4.190)$$

$$E_k = \frac{\tau^3}{N^3} LM^2 U \binom{k+1}{3} + \mathcal{O}\left(\frac{k^3 \tau^4}{N^4}\right). \quad (4.191)$$

For the induction step, we use triangle and norm inequalities to find the estimate

$$\begin{aligned} E_k &\leq \Delta_k \\ &\quad + D_{k-1} \cdot \left\| \exp\left(\frac{\tau}{N} \hat{B}_{k-2}\right) \right\| \cdots \left\| \exp\left(\frac{\tau}{N} \hat{B}_0\right) \right\| \\ &\quad + \left\| \exp\left(\frac{\tau}{N} \tilde{B}_{k-1}\right) \right\| \cdot D_{k-2} \cdot \left\| \exp\left(\frac{\tau}{N} \hat{B}_{k-3}\right) \right\| \cdots \left\| \exp\left(\frac{\tau}{N} \hat{B}_0\right) \right\| \\ &\quad + \dots \\ &\quad + \left\| \exp\left(\frac{\tau}{N} \tilde{B}_{k-1}\right) \right\| \cdots \left\| \exp\left(\frac{\tau}{N} \tilde{B}_1\right) \right\| \cdot D_0 \\ &\leq \Delta_k + (D_{k-1} + \dots + D_0) e^{(k-1)M \frac{\tau}{N}} \end{aligned} \quad (4.192)$$

from which it is already evident that (4.189) and (4.190) together imply (4.191).

Assume now that (4.189)–(4.191) have been established for all indices up to $k-1$. We have then by the Lipschitz condition on B that

$$\begin{aligned} \left\| \hat{B}_k - \tilde{B}_k \right\| &\leq L \|\hat{u}_k - \tilde{u}_k\| \\ &\leq LE_{k-1}U \end{aligned} \quad (4.193)$$

and further, using also the boundedness condition on B , that

$$\begin{aligned} D_k &\leq \left\| \exp\left(\frac{\tau}{N}(\hat{B}_k - \tilde{B}_k)\right) - I \right\| \cdot \left\| \exp\left(\frac{\tau}{N}\tilde{B}_k\right) \right\| \\ &\leq \left(\frac{\tau}{N} \cdot \frac{\tau^3}{N^3} LM^2 U^2 \binom{k}{3} + O\left(\frac{\tau^5}{N^5} \binom{k}{3}\right) \right) \left(1 + \frac{\tau M}{N} + O\left(\frac{\tau^2}{N^2}\right) \right) \end{aligned} \quad (4.194)$$

thus by virtue of $\binom{k}{3} = O(k^3)$ the estimate (4.189) for D_k .

For Δ_k we start with the observation

$$\begin{aligned} Z_k - \exp\left(\frac{\tau}{N}\hat{B}_{k-1}\right)Z_{k-1} \\ = \exp\left(\frac{\tau}{N}\sum_{j=0}^{k-1}\hat{B}_j\right) - \exp\left(\frac{\tau}{N}\hat{B}_{k-1}\right)\exp\left(\frac{\tau}{N}\sum_{j=0}^{k-2}\hat{B}_j\right) \end{aligned} \quad (4.195)$$

and use the Baker-Campbell-Hausdorff series to estimate the right-hand side:

$$\begin{aligned} Z_k - \exp\left(\frac{\tau}{N}\hat{B}_{k-1}\right)Z_{k-1} \\ = \exp\left(\frac{\tau}{N}\hat{B}_{k-1}\right)\exp\left(\frac{\tau}{N}\sum_{j=0}^{k-2}\hat{B}_j\right) \\ \cdot \left(\exp\left(\left[\frac{\tau}{N}\hat{B}_{k-1}, \frac{\tau}{N}\sum_{j=0}^{k-2}\hat{B}_j\right]\right) \left(1 + O\left(\frac{\tau}{N}\right)\right) - I \right) \end{aligned} \quad (4.196)$$

where the $O\left(\frac{\tau}{N}\right)$ contribution comes from higher order commutators in the Baker-Campbell-Hausdorff series.

For the commutator we use the decomposition

$$\left[\hat{B}_{k-1}, \sum_{j=0}^{k-2} \hat{B}_j \right] = \sum_{j=0}^{k-2} [\hat{B}_{k-1}, \hat{B}_j] = \sum_{j=0}^{k-2} [\hat{B}_{k-1} - \hat{B}_j, \hat{B}_j] \quad (4.197)$$

which can because of

$$\begin{aligned} \left\| \hat{B}_{k-1} - \hat{B}_j \right\| &= \left\| B(\hat{u}(t_{k-1})) - B(\hat{u}(t_j)) \right\| \leq L \|\hat{u}(t_{k-1}) - \hat{u}(t_j)\| \\ &\leq LMU(t_{k-1} - t_j) = LMU \frac{(k-1-j)\tau}{N} \end{aligned} \quad (4.198)$$

be used to derive the estimate

$$\begin{aligned}
 & \left\| Z_k - \exp\left(\frac{\tau}{N} \hat{B}_{k-1}\right) Z_{k-1} \right\| \\
 & \leq \left\| \exp\left(\frac{\tau}{N} \hat{B}_{k-1}\right) \exp\left(\frac{\tau}{N} \sum_{j=0}^{k-2} \hat{B}_j\right) \right\| \cdot \frac{\tau^3}{N^3} \cdot 2LM^2 U \sum_{j=0}^{k-2} (k-1-j) \\
 & \quad \cdot \left(1 + \mathcal{O}\left(\frac{\tau}{N}\right)\right) \\
 & \leq \exp\left(\frac{k\tau M}{N}\right) \cdot \frac{\tau^3}{N^3} \cdot 2LM^2 U \binom{k}{2} \cdot \left(1 + \mathcal{O}\left(\frac{\tau}{N}\right)\right). \tag{4.199}
 \end{aligned}$$

Using the induction hypothesis we obtain

$$\begin{aligned}
 \Delta_k & \leq \left\| Z_k - \exp\left(\frac{\tau}{N} \hat{B}_{k-1}\right) Z_{k-1} \right\| + \left\| \exp\left(\frac{\tau}{N} \hat{B}_{k-1}\right) \right\| \Delta_{k-1} \\
 & \leq \exp\left(\frac{k\tau M}{N}\right) \cdot \frac{\tau^3}{N^3} \cdot 2LM^2 U \binom{k}{2} \cdot \left(1 + \mathcal{O}\left(\frac{\tau}{N}\right)\right) + \exp\left(\frac{\tau M}{N}\right) \cdot \Delta_{k-1} \\
 & \leq \exp\left(\frac{k\tau M}{N}\right) \cdot \frac{\tau^3}{N^3} \cdot 2LM^2 U \cdot \left(\binom{k}{2} + \binom{k}{3}\right) \cdot \left(1 + \mathcal{O}\left(\frac{\tau}{N}\right)\right), \tag{4.200}
 \end{aligned}$$

thus the claim for Δ_k .

Since $\binom{k+1}{3} = \mathcal{O}(N^3)$, this completes the proof. \square

Convergence of time-discrete exp-int approximation to the continuous case. Finally, we will show that the time-discrete exp-int approximation (4.181) approximates its time-continuous counterpart (4.160) on the interval $[t_0, t^*]$ of length τ with an error $\mathcal{O}\left(\frac{\tau^2}{N}\right)$ that goes to zero when the number N of time steps goes to ∞ .

To this end, consider again a subdivision of $[t_0, t^*]$ according to $t_0 < t_1 < \dots < t_{N-1} < t_N = t^*$ with $t_{k+1} - t_k = \frac{\tau}{N}$, where $N \geq 2$. Consider further the solution \check{y} of

$$\check{y}(\tilde{t}) = \exp\left(\int_{t_0}^{\tilde{t}} B(\check{y}(t)) dt\right) u_0 \tag{4.201}$$

for $\tilde{t} \in [t_0, t^*]$ (compare (4.161) in Proposition 4.4. The solution of the time-discrete exp-int approximation is called \hat{u} as before.

Lemma 4.7 For the functions \hat{u} and \check{y} the estimate

$$\|\check{y}(t_N) - \hat{u}(t_N)\| = \mathcal{O}\left(\frac{\tau^2}{N}\right) \tag{4.202}$$

holds.

Proof. We show inductively the two statements

$$\|\ddot{u}(t_k) - \hat{u}(t_k)\| \leq kLMU^2 \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \quad (4.203)$$

$$\left\| \int_{t_0}^{t_k} B(\ddot{u}(t)) dt - \frac{\tau}{N} \sum_{j=0}^{k-1} B(\hat{u}(t_j)) \right\| \leq kLMU \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right). \quad (4.204)$$

The base case $k = 0$ is clear, as the left-hand sides of both inequalities vanish. Assume therefore that (4.203), (4.204) are proven for index $k - 1$. For $t \in [t_{k-1}, t_k]$ we have then, using the estimate $\|\dot{u}\| \leq \|B\| \|u_0\| \leq MU$ and the induction hypothesis for (4.203),

$$\begin{aligned} \|\ddot{u}(t) - \hat{u}(t_{k-1})\| &\leq \|\ddot{u}(t) - \ddot{u}(t_{k-1})\| + \|\ddot{u}(t_{k-1}) - \hat{u}(t_{k-1})\| \\ &\leq MU \frac{\tau}{N} + (k-1)LMU^2 \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \\ &= MU \frac{\tau}{N} \left(1 + (k-1)LU \frac{\tau}{N}\right) + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right). \end{aligned} \quad (4.205)$$

By the Lipschitz condition on B this implies

$$\|B(\ddot{u}(t)) - B(\hat{u}(t_{k-1}))\| \leq LMU \frac{\tau}{N} \left(1 + (k-1)LU \frac{\tau}{N}\right) + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \quad (4.206)$$

and after integration over $[t_{k-1}, t_k]$

$$\begin{aligned} &\left\| \int_{t_{k-1}}^{t_k} B(\ddot{u}(t)) dt - \frac{\tau}{N} B(\hat{u}(t_{k-1})) \right\| \\ &\leq LMU \frac{\tau^2}{N^2} \left(1 + (k-1)LU \frac{\tau}{N}\right) + \mathcal{O}\left(\frac{k^2\tau^4}{N^4}\right). \end{aligned} \quad (4.207)$$

With the induction hypothesis for (4.204) and the triangle inequality for the norms on the l.h.s. it follows that

$$\begin{aligned} \left\| \int_{t_0}^{t_k} B(\ddot{u}(t)) dt - \frac{\tau}{N} \sum_{j=0}^{k-1} B(\hat{u}(t_j)) \right\| &\leq LMU \frac{\tau^2}{N^2} \left(1 + (k-1)LU \frac{\tau}{N}\right) + \mathcal{O}\left(\frac{k^2\tau^4}{N^4}\right) \\ &\quad + LMU \frac{\tau^2}{N^2} (k-1) + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) \\ &= kLMU \frac{\tau^2}{N^2} + \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right), \end{aligned} \quad (4.208)$$

i.e. the claim (4.204) for index k . Note that in the last step we used $k = \mathcal{O}(N)$, thus

$$\mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) + \mathcal{O}\left(\frac{k^2\tau^4}{N^4}\right) = \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right) + \mathcal{O}\left(\frac{k\tau^4}{N^3}\right) = \mathcal{O}\left(\frac{k^2\tau^3}{N^3}\right), \quad (4.209)$$

to combine the higher order terms.

By the exponential series it follows further that

$$\begin{aligned} & \left\| \exp\left(\int_{t_0}^{t_k} B(\check{y}(t)) dt\right) - \exp\left(\frac{\tau}{N} \sum_{j=0}^{k-1} B(\hat{y}(t_j))\right) \right\| \\ & \leq kLMU \frac{\tau^2}{N^2} + O\left(\frac{k^2 \tau^3}{N^3}\right) + O\left(\frac{k^2 \tau^4}{N^4}\right). \end{aligned} \quad (4.210)$$

The higher order terms on the right-hand side can again be combined via (4.209). Subtracting (4.181) from (4.161) and using (4.210), $\|u_0\| \leq U$ implies the claim (4.203) which completes the induction step.

Inserting $k = N$ into (4.203) yields the claim of the lemma. \square

From the lemma it follows immediately that also the time-discrete exp-int approximation (4.181) converges to (4.160) over $[t_0, t^*]$ when the subdivision is refined infinitely, $N \rightarrow \infty$.

Combining the results. Proposition 4.4 is a straightforward concatenation of the estimates from Lemmas 4.5, 4.6 and 4.7.

Chapter 5

Spatially Discrete Analysis of Structure-Enhancing Processes

In this chapter, we are concerned with another group of PDE-based image filters for which the semidiscrete analysis via dynamical systems has proven fruitful. Our study will include one type of shock filters and two types of inverse diffusion filters. These filters have in common that they are designed to enhance discontinuous structures like edges, but their theoretical understanding on the continuous level suffers considerably from ill-posedness. The focus of our investigation here is in all cases on a theoretical analysis of semidiscrete and in part fully discrete formulations. Also, the results obtained so far refer to the 1-D case. The study of the 2-D analoga is part of ongoing research.

Shock filters. While shock filters are popular morphological image enhancement methods, no well-posedness theory is available for their corresponding partial differential equations (PDEs). By analysing the dynamical system that results from a space discretisation of a 1-D shock filter PDE, we derive an analytical solution and prove well-posedness. We show that the results carry over to the fully discrete case when an explicit time discretisation is applied. Finally we establish an equivalence result between discrete shock filtering and local mode filtering. The results on shock filters have been published first in [224, 226]. The presentation here follows mainly [226]. The relation between shock filters and mode filtering described in 5.1.3 has been contributed by work of I. Galić.

Stabilised inverse linear diffusion. We consider a semidiscrete model problem for the approximation of stabilised inverse linear diffusion processes. It is known that numerical realisations of this and other processes involve the so-called staircasing phenomenon: when sharpening monotone data profiles, fully discrete methods

generally introduce stepfunction-type solutions reminiscent of staircases. By analysing dynamical systems, we show that staircasing is that already the semidiscrete numerical model contains the relevant information on the occurrence of staircasing. Numerical experiments confirm and complement the theoretical findings. These results have been achieved in collaboration with M. Breuß and published in [38]. Our presentation here follows closely this reference.

Forward-and-backward diffusion. Finally, we will consider a nonlinear diffusion process in which the diffusivity $g = g(|\nabla u|^2)$ can even attain negative values. We demand only that $g(0) =: c_1$ is positive, g is positive for a small interval $[0, z)$, $z > 0$, and g is bounded from below by a negative constant c_2 with $|c_2| < c_1$. This process has been studied, among others, by G. Gilboa et al., under the name *forward-and-backward (FAB) diffusion* [95]. It relates to triple-well-potentials discussed by the same authors in [96]. In analysing FAB diffusion, we restrict ourselves to the fully discrete situation.

5.1 Shock Filters in 1D

Shock filters are morphological image enhancement methods where dilation is performed around maxima and erosion around minima. Iterating this process leads to a segmentation with piecewise constant segments that are separated by discontinuities, so-called shocks. This makes shock filtering attractive for a number of applications where edge sharpening and a piecewise constant segmentation is desired.

In 1975 the first shock filters have been formulated by Kramer and Bruckner in a fully discrete manner [125], while first continuous formulations by means of partial differential equations (PDEs) have been developed in 1990 by Osher and Rudin [155]. The relation of these methods to the discrete Kramer–Bruckner filter became clear several years later [100, 180]. PDE-based shock filters have been investigated in a number of papers. Many of them proposed modifications with higher robustness under noise [5, 94, 123, 156, 180], but also coherence-enhancing shock filters [209] and numerical schemes have been studied [170].

Let us consider some continuous d -dimensional initial image $f : \mathbb{R}^d \rightarrow \mathbb{R}$. In the simplest case of a PDE-based shock filter [155], one obtains a filtered version $u(x, t)$ of $f(x)$ by solving the evolution equation

$$\partial_t u = -\operatorname{sgn}(\Delta u) |\nabla u| \quad (t \geq 0) \quad (5.1)$$

with f as initial condition, i.e., $u(x, 0) = f(x)$. Experimentally one observes that within finite “evolution time” t , a piecewise constant, segmentation-like result is obtained (see Fig. 5.1).

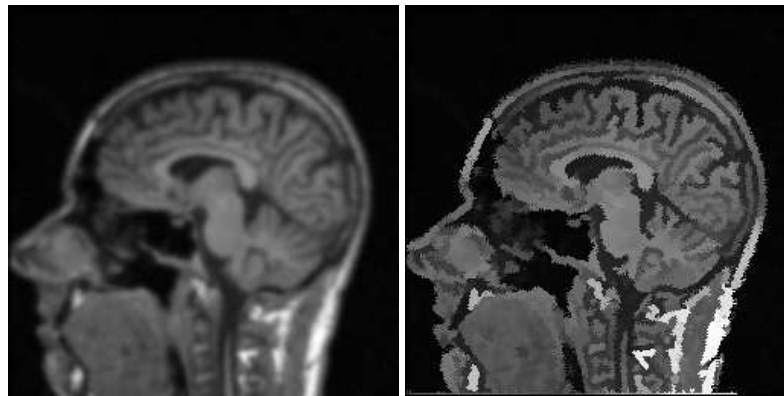


Figure 5.1: *Left: (a)* Original image (blurred version of Figure 1.1(b)). *Right: (b)* After applying the Osher–Rudin shock filter. – Following [224].

Specialising to the one-dimensional case, we obtain

$$\partial_t u = -\operatorname{sgn}(\partial_{xx} u) |\partial_x u| = \begin{cases} |\partial_x u|, & \partial_{xx} u < 0, \\ -|\partial_x u|, & \partial_{xx} u > 0, \\ 0, & \partial_{xx} u = 0. \end{cases} \quad (5.2)$$

It is clearly visible that this filter performs dilation $\partial_t u = |\partial_x u|$ in concave segments of u , while in convex parts the erosion process $\partial_t u = -|\partial_x u|$ takes place. The time t specifies the radius of the interval (a 1-D disk) $[-t, t]$ as structuring element. For a derivation of these PDE formulations for classical morphological operations, see e.g. [39].

While there is clear experimental evidence that shock filtering is a useful operation, no analytical solutions and well-posedness results are available for PDE-based shock filters. In general this problem is considered to be too difficult, since shock filters have some connections to classical ill-posed problems such as backward diffusion [155, 156].

The main goal of this section is to show that it is possible to establish analytical solutions and well-posedness as soon as we study the *semidiscrete* case with a spatial discretisation and a continuous time parameter t . This case is of great practical relevance, since digital images already induce a natural space discretisation. For the sake of simplicity we restrict ourselves to the 1-D case. We also show that these results carry over to the fully discrete case with an explicit (Euler forward) time discretisation, and we establish an equivalence result between shock filtering and a specific image enhancement method called mode filtering.

In 5.1.1 we present an analytical solution and a well-posedness proof for the semidiscrete case, whereas corresponding fully discrete results are given in 5.1.2. An equivalence result between shock and mode filters is presented in 5.1.3.

5.1.1 Semidiscrete Case

5.1.1.1 Problem Statement

In this section, we are concerned with a spatial discretisation of (5.2) which we will describe now. The time variable remains continuous here. Throughout the following, the notion *semidiscrete* will refer to this setting.

Problem. Let $(\dots, u_0(t), u_1(t), u_2(t), \dots)$ be a time-dependent bounded real-valued signal which evolves according to

$$\dot{u}_i = \begin{cases} \max(u_{i+1} - u_i, u_{i-1} - u_i, 0), & 2u_i > u_{i+1} + u_{i-1}, \\ \min(u_{i+1} - u_i, u_{i-1} - u_i, 0), & 2u_i < u_{i+1} + u_{i-1}, \\ 0, & 2u_i = u_{i+1} + u_{i-1} \end{cases} \quad (5.3)$$

with the initial conditions

$$u_i(0) = f_i. \quad (5.4)$$

Assume further that the signal is either of infinite length with compact support, or finite with reflecting boundary conditions.

Here, \dot{u}_i denotes the time derivative of $u_i(t)$. Like (5.2), this filter switches between dilation and erosion depending on the local convexity or concavity of the signal. Dilation and erosion themselves are modeled by upwind-type discretisations [158], which are well established for dilation and erosion PDEs because they guarantee the detection of local extrema and stabilise the discretised process by adapting the discrete representation to the local directedness of the PDE evolution.

It is important that in case $2u_i > u_{i+1} + u_{i-1}$ the two neighbour differences $u_{i+1} - u_i$ and $u_{i-1} - u_i$ cannot be simultaneously positive; with the opposite inequality they cannot be simultaneously negative. In fact, whenever the maximum or minimum in (5.3) does not select its third argument, zero, it returns the absolutely smaller of the neighbour differences.

No modification of (5.3) is needed for finite-length signals with reflecting boundary conditions. In this case, each boundary pixel has one vanishing neighbour difference.

In order to study the solution behaviour of this system, we have to specify the possible solutions, taking into account that the right-hand side of (5.3) may involve discontinuities. Similar to the conditions for solutions of 1D TV flow formulated on page 103, we call a time-dependent signal $u(t) = (\dots, u_1(t), u_2(t), u_3(t), \dots)$ a *solution* of (5.3) if

- (I) each u_i is a continuous, piecewise differentiable function of t ,

- (II) each u_i satisfies (5.3) for all times t for which $\dot{u}_i(t)$ exists,
- (III) for $t = 0$, the right-sided derivative $\dot{u}_i^+(0)$ equals the right-hand side of (5.3) if $2u_i(0) \neq u_{i+1}(0) + u_{i-1}(0)$.

Remark. Let us point out that extremality of pixels is handled strictly local: A pixel u_i of a 1-D space-discrete signal u is called a (discrete) local extremum whenever $(u_i - u_{i-1})(u_{i+1} - u_i) \leq 0$. For example, in the sequence of five pixels $u_1 > u_2 = u_3 = u_4 > u_5$, pixels u_2 and u_3 are local minima while u_3 and u_4 are local maxima.

5.1.1.2 Well-Posedness Results

The following theorem contains our main result.

Theorem 5.1 (Well-Posedness) *For our Problem, assume that the equality $f_{k+1} - 2f_k + f_{k-1} = 0$ does not hold for any pixel f_k which is not a local maximum or minimum of f . Then the following are true:*

- (i) *Existence and uniqueness:* The Problem has a unique solution for all $t \geq 0$.
- (ii) *Maximum–minimum principle:* If there are real bounds a, b such that $a < f_k < b$ holds for all k , then $a < u_k(t) < b$ holds for all k and all $t \geq 0$.
- (iii) *l_∞ -stability:* There exists a $\delta > 0$ such that for any initial signal \tilde{f} with $\|\tilde{f} - f\|_\infty < \delta$ the corresponding solution \tilde{u} satisfies the inequality

$$\|\tilde{u}(t) - u(t)\|_\infty < \|\tilde{f} - f\|_\infty$$

for all $t > 0$. The solution therefore depends l_∞ -continuously on the initial conditions within a neighbourhood of f .

- (iv) *Total variation preservation:* If the total variation of f is finite, then the total variation of u at any time $t \geq 0$ equals that of f .
- (v) *Steady state:* For $t \rightarrow \infty$, the signal u converges to a piecewise constant signal. The jumps in this signal are located at the steepest slope positions of the original signal.

All statements of this theorem follow from an explicit analytical solution of the Problem that will be described in the following proposition.

Proposition 5.2 (Analytical Solution) *For our Problem, let the segment (f_1, \dots, f_m) be strictly decreasing and concave in all pixels. Assume that the leading pixel f_1 is either a local maximum or a neighbour to a convex pixel $f_0 > f_1$. Then the following hold for all $t \geq 0$:*

- (i) If f_1 is a local maximum of f , then $u_1(t)$ is a local maximum of $u(t)$.
- (ii) If f_1 is neighbour to a convex pixel $f_0 > f_1$, then $u_1(t)$ also has a convex neighbour pixel $u_0(t) > u_1(t)$.
- (iii) The segment (u_1, \dots, u_m) remains strictly decreasing and concave in all pixels. The grey values of all pixels at time t are given by

$$u_k(t) = C \cdot \left(1 + (-1)^k e^{-2t} - e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} (1 + (-1)^{k-j}) \right) + e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} f_{k-j} - (-1)^k f_1 e^{-t} \left(e^{-t} - \sum_{j=0}^{k-2} \frac{(-t)^j}{j!} \right) \quad (5.5)$$

for $k = 1, \dots, m$, where

$$C = \begin{cases} f_1(0) & \text{if } f_1 \text{ is a local maximum of } f, \\ \frac{1}{2}(f_0(0) + f_1(0)) & \text{otherwise.} \end{cases} \quad (5.6)$$

- (iv) At no time $t \geq 0$, the equation $2u_i(t) = u_{i+1}(t) + u_{i-1}(t)$ becomes true for any $i \in \{1, \dots, m\}$.

Analogous statements hold for increasing concave and for convex signal segments.

In a signal that contains no locally linear pixels (such with $2f_i = f_{i+1} + f_{i-1}$), each pixel belongs to a chain of either concave or convex pixels led by an extremal pixel or an “inflection pair” of a convex and a concave pixel. Therefore Proposition 5.2 completely describes the dynamics of such a signal. Let us prove this proposition.

Proof. We show in steps 1.–3. that the claimed evolution equations hold as long as the initial monotonicity and convexity properties of the signal segment prevail. Step 4. then completes the proof by demonstrating that the evolution equations preserve exactly these monotonicity and convexity requirements.

1. From (5.3) it is clear that any pixel u_i which is extremal at time t has $\dot{u}_i(t) = 0$ and therefore does not move. Particularly, if f_1 is a local maximum of f , then $u_1(t)$ remains constant as long as it continues to be a maximum.

2. If $u_0 > u_1$, u_0 is convex and u_1 concave for $t \in [0, T)$. Then we have for these pixels

$$\begin{aligned} \dot{u}_0 &= u_1 - u_0, \\ \dot{u}_1 &= u_0 - u_1, \end{aligned} \quad (5.7)$$

which by the substitutions $y := \frac{1}{2}(u_0 + u_1)$ and $v := u_1 - u_0$ becomes

$$\begin{aligned} \dot{y} &= 0, \\ \dot{v} &= -2v. \end{aligned} \quad (5.8)$$

This system of linear ordinary differential equations (ODEs) has the solution $y(t) = y(0) = C$ and $v(t) = v(0)\exp(-2t)$. Backsubstitution gives

$$\begin{aligned} u_0(t) &= C \cdot (1 - e^{-2t}) + f_0 e^{-2t}, \\ u_1(t) &= C \cdot (1 - e^{-2t}) + f_1 e^{-2t}. \end{aligned} \quad (5.9)$$

This explicit solution is valid as long as the convexity and monotonicity properties of u_0 and u_1 do not change.

3. Assume the monotonicity and convexity conditions required by the proposition for the initial signal hold for $u(t)$ for all $t \in [0, T)$. Then we have in all cases, defining C as in the proposition, the system of ODEs

$$\begin{aligned} \dot{u}_1 &= -2(u_1 - C), \\ \dot{u}_k &= u_{k-1} - u_k, \quad k = 2, \dots, m \end{aligned} \quad (5.10)$$

for $t \in [0, T)$. We substitute further $v_k := u_k - C$ for $k = 1, \dots, m$ as well as $w_1 := v_1$ and $w_k := v_k + (-1)^k v_1$ for $k = 2, \dots, m$. This leads to the system

$$\begin{aligned} \dot{w}_1 &= -2w_1, \\ \dot{w}_2 &= -w_2, \\ \dot{w}_k &= w_{k-1} - w_k, \quad k = 3, \dots, m. \end{aligned} \quad (5.11)$$

This system of linear ODEs has the unique solution

$$\begin{aligned} w_1(t) &= w_1(0)e^{-2t}, \\ w_k(t) &= e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} w_{k-j}(0), \quad k = 2, \dots, m \end{aligned} \quad (5.12)$$

which after reverse substitution yields (5.5) for all $t \in [0, T]$.

4. Note that (5.7) and (5.10) are systems of linear ODEs which have the unique explicit solutions (5.9) and (5.5) for all $t > 0$. As long as the initial monotonicity and convexity conditions are satisfied, the solutions of (5.3) coincide with those of the linear ODE systems.

We prove therefore that the solution (5.5) fulfils the monotonicity condition

$$u_k(t) - u_{k-1}(t) < 0, \quad k = 2, \dots, m, \quad (5.13)$$

and the concavity conditions

$$u_{k+1}(t) - 2u_k(t) + u_{k-1}(t) < 0, \quad k = 1, \dots, m, \quad (5.14)$$

for all $t > 0$ if they are valid for $t = 0$. To see this, we calculate first

$$\begin{aligned} u_k(t) - u_{k-1}(t) &= e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} (f_{k-j} - f_{k-1-j}) \\ &\quad + 2e^{-t} (-1)^{k-1} \left(e^{-t} - \sum_{j=0}^{k-2} \frac{(-t)^j}{j!} \right) (f_1 - C). \end{aligned} \quad (5.15)$$

By hypothesis, $f_{k-j} - f_{k-1-j}$ and $f_1 - C$ are negative. Further, the difference expression $\exp(-t) - \sum_{j=0}^{k-2} (-t)^j / j!$ is just the error of the (alternating) Taylor series of $\exp(-t)$, thus having the same sign $(-1)^{k-1}$ as the first neglected member. Consequently, monotonicity is preserved by (5.5) for all $t > 0$.

Second, we have for $k = 2, \dots, m-1$

$$\begin{aligned} u_{k+1}(t) - 2u_k(t) + u_{k-1}(t) &= e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} (f_{k-j+1} - 2f_{k-j} + f_{k-j-1}) \\ &\quad + 4e^{-t} (-1)^k \left(e^{-t} - \sum_{j=0}^{k-1} \frac{(-t)^j}{j!} \right) (f_1 - C) \quad (5.16) \\ &\quad + e^{-t} \frac{t^{k-1}}{(k-1)!} (f_2 - 3f_1 + 2C) \end{aligned}$$

which is seen to be negative by similar reasoning as above.

Concavity at $u_m(t)$ follows in nearly the same way. By extending (5.5) to $k = m+1$, one obtains not necessarily the true evolution of u_{m+1} since that pixel is not assumed to be included in the concave segment. However, the true trajectory of u_{m+1} can only lie below or on that predicted by (5.5).

Third, if f_1 is a maximum of f , then $u_1(t)$ remains one for all $t > 0$ which also ensures concavity at u_1 . If f_1 has a convex neighbour pixel $f_0 > f_1$, we have instead

$$u_2(t) - 2u_1(t) + u_0(t) = e^{-t} (f_2 - 2f_1 + f_0) + 4e^{-t} (1 - e^{-t}) (f_1 - C) < 0 \quad (5.17)$$

which is again negative for all $t > 0$.

Finally, we remark that the solution (5.9) ensures $u_0(t) > u_1(t)$ for all $t > 0$ if it holds for $t = 0$. That convexity at u_0 is preserved can be established by analogous reasoning as for the concavity at u_1 .

Since the solutions from the linear systems guarantee preservation of all monotonicity and convexity properties which initially hold for the considered segment, these solutions are the solutions of (5.3) for all $t > 0$. \square

Proof of Theorem 5.1. Existence and uniqueness of the solution follow from the proof of Proposition 5.2.

The maximum–minimum principle and preservation of total variation follow from the fact that extrema do not move, and monotonicity is preserved for all $t > 0$. Note that by the specification of our Problem, each non-extremal pixel in the signal belongs to a segment enclosed by two extrema.

For the l_∞ -stability, note that for each admissible initial signal f , there exists a lower bound $\gamma > 0$ for all values of $|f_{k-1} - f_k|$ which are not zero, and a lower bound $\eta > 0$ for all values of $|f_{k-1} - 2f_k + f_{k+1}|$ for pixels k which are not local extrema of f .

Let a signal \tilde{f} with $\|\tilde{f} - f\|_\infty =: d < \min\{\gamma/2, \eta/4\}$ be given. One easily checks that then the monotonicity and convexity properties of all strictly monotone segments of f are preserved in \tilde{f} . Moreover, isolated extremal pixels of f will be such in \tilde{f} . Only chains of equal pixels $f_k = \dots = f_{k+l}$ may break up in \tilde{f} , but in this case the corresponding chain $\tilde{f}_k, \dots, \tilde{f}_{k+l}$ contains at least one extremum or one inflection pair. To sum up, each monotone and convex/concave segment in f with one of the pixels $f_k = \dots = f_{k+l}$ as leading pixel is turned into a segment of equal character in \tilde{f} whose leading pixel is one of $\tilde{f}_k, \dots, \tilde{f}_{k+l}$.

Let us choose without loss of generality a pixel \tilde{f}_k within a decreasing and concave segment as in the proposition. By virtue of $|\tilde{C} - C| \leq d$ and $|\tilde{f}_j - f_j| \leq d$, we can estimate the difference of the explicit solutions (5.5) for \tilde{u}_k and u_k :

$$\begin{aligned} |\tilde{u}_k(t) - u_k(t)| &\leq d \cdot \left(\left| 1 - e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} + (-1)^k e^{-t} \left(e^{-t} - \sum_{j=0}^{k-2} \frac{(-t)^j}{j!} \right) \right| \right. \\ &\quad \left. + e^{-t} \sum_{j=0}^{k-2} \frac{t^j}{j!} + e^{-t} \cdot \left| e^{-t} - \sum_{j=0}^{k-2} \frac{(-t)^j}{j!} \right| \right) \\ &\leq d \end{aligned} \quad (5.18)$$

which proves the l_∞ -stability statement with $\delta := \min\{\gamma/2, \eta/4\}$.

Finally, the convergence to a steady state is obvious from the exponential decay of all summands but C in (5.5). \square

Remark. The fact that the signal reaches its steady state only for $t \rightarrow \infty$ stands in contrast to the behaviour of space-continuous dilation and erosion where extrema propagate in space with constant speed. In our semidiscrete setting, non-extremal pixels only asymptotically approach their limit values. This can be seen as a blurring, or approximation error, which is the price for the performed spatial discretisation.

5.1.1.3 Signals With Non-Extremal Linear Pixels

Our well-posedness statements in the previous section explicitly exclude those signals which contain pixels f_k with $f_{k+1} - 2f_k + f_{k-1} = 0$. For brevity, we shall call such pixels *non-extremal linear pixels*.

As we are going to demonstrate, no uniqueness and continuous dependence on initial condition holds for initial signals f with non-extremal linear pixels. First of all,

condition (III) allows the right-sided time derivative of such a pixel at $t = 0$ to deviate from the right-hand side of equation (5.3). Without this relaxation, the condition would in many cases stand in contradiction to the uniquely determined evolution of non-linear surrounding pixels, preventing the existence of solutions.

While the evolution of the neighbours of a non-extremal linear pixel will often force it to become convex or concave, this process is never unique: Instead, one can label each non-extremal linear pixel at $t = 0$ *arbitrarily* to impose either convexity or concavity. Whatever labelling is chosen, it leads to a consistent evolution for $t > 0$ in which no linear pixels reappear. This is precised in the following proposition.

Proposition 5.3 (Forking Solutions at Non-Extremal Linear Pixels) *Let an initial signal $f = (\dots, f_1, f_2, f_3, \dots)$ be given. For each non-extremal linear pixel f_k , let a sign $\sigma_k \in \{+1, -1\}$ be chosen. Then there exists a unique solution of (5.3) in the sense of (I)–(III) with initial conditions (5.4) such that for each non-extremal linear pixel f_k , the right-sided derivative of u_k at $t = 0$ is*

$$\dot{u}_k^+(0) = \begin{cases} \max(u_{i+1} - u_i, u_{i-1} - u_i, 0), & \sigma_k = +1, \\ \min(u_{i+1} - u_i, u_{i-1} - u_i, 0), & \sigma_k = -1. \end{cases} \quad (5.19)$$

For none of these solutions, non-extremal linear pixels exist in $u(t)$ for $t > 0$.

Remark. As there is no “natural” labelling of linear pixels as convex or concave, this proposition expresses that for initial signals with non-extremal linear pixels, the spatially discrete shock filter is highly nonunique. Moreover, for such initial signals the solutions no longer depend continuously on the initial conditions: Considering a non-extremal linear pixel f_k in a signal f , it is clear that every neighbourhood of f will contain signals \tilde{f} in which \tilde{f}_k is strictly convex, and such in which it is strictly concave.

Proof. Assume we are given a fixed choice of the σ_k . Interpreting non-extremal linear pixels with $\sigma_k = +1$ as convex and such with $\sigma_k = -1$ as concave, we have again a segmentation of the entire signal into concave and convex regions as for signals without non-extremal linear pixels. For each of these regions, we proceed as in the proof of Proposition 5.2 by rewriting the evolution into a system of linear ODEs which has verbatim the same analytical solutions as before.

Following further the proof of Proposition 5.2, one finds first that the proof of monotonicity preservation suffers no change at all. In the concavity preservation proof one finds that some of the summands of type $(f_{j-1} - 2f_j + f_{j+1})$ in the inequalities used there for $u_{k+1}(t) - 2u_k(t) + u_{k+1}(t)$ now vanish. However, for the entire right-hand side to vanish it is necessary that $(f_1 - C)$ and all $(f_{j-1} - 2f_j + f_{j+1})$ for $j = 1, \dots, k$ are zero. This is true if and only if f_k itself is extremal. Consequently, under the

evolution of the linear system, non-extremal pixels which are linear at $t = 0$ become convex for all positive t if $\sigma_k = +1$, or concave for all positive t if $\sigma_k = -1$. \square

Remark. In some cases, there can even exist more solutions than listed in Proposition 5.3, in particular such in which non-extremal linear pixels retain this property at positive evolution times. We sketch just one simple case: Assume that a non-extremal linear pixel f_k is surrounded by a concave and a convex segment in the sense of Proposition 5.2, and that these segments are completely point-symmetric with respect to (k, f_k) . Then there will also exist possible evolutions with $u_k(t) = f_k$ for all $t > 0$. We will, however, not give a full classification of these solutions here since our primary goal to give necessary and sufficient conditions for well-posedness has already been achieved with Proposition 5.3.

As to the solution concept, one could also think of introducing a way to deal with the instability at linear pixels would be to introduce a degree of freedom into the right-hand side of (5.3) for non-extremal linear pixels, very much in the same spirit as the treatment of $\text{sgn}(0)$ in the analysis of TV diffusion [44]. However, since this would still not remove the ambiguities involved here, it seems not too sensible to do so.

5.1.2 Fully Discrete Case

In the following we discuss time discretisations of our time-continuous system. As usual, we denote the time step by $\tau > 0$.

5.1.2.1 Explicit Time Discretisation

A straightforward explicit time discretisation of our Problem reads as follows:

Time-Discrete Problem. Let $(\dots, u_0^l, u_1^l, u_2^l, \dots)$, $l = 0, 1, 2, \dots$ be a series of bounded real-valued signals which satisfy the equations

$$\frac{u_i^{l+1} - u_i^l}{\tau} = \begin{cases} \max(u_{i+1}^l - u_i^l, u_{i-1}^l - u_i^l, 0), & 2u_i^l > u_{i+1}^l + u_{i-1}^l, \\ \min(u_{i+1}^l - u_i^l, u_{i-1}^l - u_i^l, 0), & 2u_i^l < u_{i+1}^l + u_{i-1}^l, \\ 0, & 2u_i^l = u_{i+1}^l + u_{i-1}^l \end{cases} \quad (5.20)$$

with the initial conditions

$$u_i^0 = f_i. \quad (5.21)$$

Assume further that the signal is either of infinite length or finite with reflecting boundary conditions.

Theorem 5.4 (Time-Discrete Well-Posedness) *Assume that in the Time-Discrete Problem the equality $f_{k+1} - 2f_k + f_{k-1} = 0$ does not hold for any pixel f_k which is not a local maximum or minimum of f . Assume further that $\tau < 1/2$. Then the statements of Theorem 5.1 are valid for the solution of the Time-Discrete Problem if only $u_k(t)$ for $t > 0$ is replaced everywhere by u_k^l with $l = 0, 1, 2, \dots$*

The existence and uniqueness of the solution of the Time-Discrete Problem for $l = 0, 1, 2, \dots$ is obvious. Maximum–minimum principle, l_∞ -stability, total variation preservation and the steady state property are immediate consequences of the following proposition. It states that for $\tau < 1/2$ all qualitative properties of the time-continuous solution transfer to the time-discrete case.

Proposition 5.5 (Time-Discrete Solution) *Let u_i^l be the value of pixel i in time step l of the solution of our Time-Discrete Problem with time step size $\tau < 1/2$. Then the following hold for all $l = 0, 1, 2, \dots$:*

- (i) *If u_1^l is a local maximum of u^l , then u_1^{l+1} is a local maximum of u^{l+1} .*
- (ii) *If u_1^l is a concave pixel neighbouring to a convex pixel $u_0^l > u_1^l$, then u_1^{l+1} is again concave and has a convex neighbour pixel $u_0^{l+1} > u_1^{l+1}$.*
- (iii) *If the segment (u_1^l, \dots, u_m^l) is strictly decreasing and concave in all pixels, and u_1^l is either a local maximum of u^l or neighbours to a convex pixel $u_0^l > u_1^l$, then the segment $(u_1^{l+1}, \dots, u_m^{l+1})$ is strictly decreasing.*
- (iv) *Under the same assumptions as in (iii), the segment $(u_1^{l+1}, \dots, u_m^{l+1})$ is strictly concave in all pixels.*
- (v) *If $2u_i^l = u_{i+1}^l + u_{i-1}^l$ holds for no pixel i , then $2u_i^{l+1} = u_{i+1}^{l+1} + u_{i-1}^{l+1}$ also holds for no pixel i .*
- (vi) *Under the assumptions of (iii), all pixels in the range $i \in \{1, \dots, m\}$ have the same limit $\lim_{l \rightarrow \infty} u_i^l = C$ with*

$$C := \begin{cases} f_1 & \text{if } f_1 \text{ is a local maximum,} \\ \frac{1}{2}(f_0 + f_1) & \text{if } f_1 \text{ neighbours to the convex pixel } f_0. \end{cases} \quad (5.22)$$

Analogous statements hold for local minima, for increasing concave and for convex signal segments.

Proof. Assume first that u_1^l is a local maximum of u^l . From the evolution equation (5.20) it is clear that $u_j^{l+1} \leq u_j^l + \tau(u_1^l - u_j^l)$ for $j = 0, 2$. For $\tau < 1$ this entails $u_j^{l+1} < u_1^l = u_1^{l+1}$, thus (i).

If instead u_i^l is a concave neighbour of a convex pixel $u_0^l > u_1^l$, then we have $u_1^{l+1} = u_1^l + \tau(u_0^l - u_1^l)$ and $u_0^{l+1} = u_0^l + \tau(u_1^l - u_0^l)$. Obviously, $u_0^{l+1} > u_1^{l+1}$ holds if and only if $\tau < 1/2$. For concavity, note that $u_2^{l+1} \leq u_2^l + \tau(u_1^l - u_2^l)$ and therefore $u_0^{l+1} - 2u_1^{l+1} + u_2^{l+1} \leq (1-\tau)(u_0^l - 2u_1^l + u_2^l) + 2\tau(u_1^l - u_0^l)$. The right-hand side is certainly negative for $\tau \leq 1/2$. An analogous argument secures convexity at pixel 0 which completes the proof of (ii).

In both cases we have $u_1^{l+1} \geq u_1^l$. Under the assumptions of (iii), (iv) we then have $u_k^{l+1} = u_k^l + \tau(u_{k-1}^l - u_k^l)$ for $k = 2, \dots, m$. If $\tau < 1$, it follows that $u_k^l < u_k^{l+1} \leq u_{k-1}^l$ for $k = 2, \dots, m$ which together with $u_1^{l+1} \geq u_1^l$ implies that $u_{k-1}^{l+1} > u_k^l$ for $k = 2, \dots, m$ and therefore (iii).

For the concavity condition we compute

$$u_{k-1}^{l+1} - 2u_k^{l+1} + u_{k+1}^{l+1} = (1-\tau)(u_{k-1}^l - 2u_k^l + u_{k+1}^l) + \tau(u_{k-2}^l - 2u_{k-1}^l + u_k^l) \quad (5.23)$$

for $k = 3, \dots, m-1$. The right-hand side is certainly negative for $\tau \leq 1$ which secures concavity in the pixels $k = 3, \dots, m-1$. Concavity in pixel m for $\tau \leq 1$ follows from essentially the same argument. However, the equation is now replaced by an inequality since for pixel $m+1$ we know only that $u_{m+1}^{l+1} \leq u_{m+1}^l + \tau(u_m^l - u_{m+1}^l)$. If u_1^l is a local maximum and therefore $u_1^{l+1} = u_1^l$, we find for pixel 2 that $u_1^{l+1} - 2u_2^{l+1} + u_3^{l+1} = (1-\tau)(u_1^l - 2u_2^l + u_3^l) + \tau(u_2^l - u_1^l)$ which again secures concavity for $\tau \leq 1$. As was proven above, concavity in pixel 1 is preserved for $\tau \leq 1/2$ such that (iv) is proven.

Under the hypothesis of (v), the evolution of all pixels in the signal is described by statements (i)–(iv) or their obvious analoga for increasing and convex segments. The claim of (v) then is obvious.

Finally, addition of the equalities $C - u_1^{l+1} = (1-2\tau)(C - u_1^l)$ and $u_{i-1}^{l+1} - u_i^{l+1} = (1-\tau)(u_{i-1}^l - u_i^l)$ for $i = 2, \dots, m$ implies that

$$C - u_k^{l+1} = (1-\tau)(C - u_k^l) - \tau(C - u_1^l) < (1-\tau)(C - u_k^l) \quad (5.24)$$

for all $k = 1, \dots, m$. By induction, we have

$$C - u_k^{l+l'} \leq (1-\tau)^{l'}(C - u_k^l) \quad (5.25)$$

where the right-hand side tends to zero for $l' \rightarrow \infty$. Together with the monotonicity preservation for $\tau < 1/2$, statement (vi) follows. \square

We remark that in the presence of non-extremal linear pixels, uniqueness fails in a similar way as in the semidiscrete setting.

5.1.2.2 Modified Explicit Time Discretisation

A closer look at the proof of Proposition 5.5 reveals that the limitation $\tau < 1/2$ is made necessary only by the situation of case (ii) of the proposition, i.e., a concave pixel following a convex one within a decreasing segment, or symmetrical situations. In the absence of such a configuration, the statements hold even for all $\tau < 1$.

By a small adaptation of the time-discrete evolution rule we can therefore obtain a scheme which satisfies well-posedness properties for time step sizes up to 1.

Modified Time-Discrete Problem. Let $(\dots, u_0^l, u_1^l, u_2^l, \dots)$, $l = 0, 1, 2, \dots$ be a series of bounded real-valued signals which is generated by the equations

$$\frac{\tilde{u}_i^l - u_i^l}{\tau} = \begin{cases} \max(u_{i+1}^l - u_i^l, u_{i-1}^l - u_i^l, 0), & 2u_i^l > u_{i+1}^l + u_{i-1}^l, \\ \min(u_{i+1}^l - u_i^l, u_{i-1}^l - u_i^l, 0), & 2u_i^l < u_{i+1}^l + u_{i-1}^l, \\ 0, & 2u_i^l = u_{i+1}^l + u_{i-1}^l, \end{cases} \quad (5.26)$$

$$u_i^{l+1} = \begin{cases} \frac{1}{2}(\tilde{u}_{i+1}^l + \tilde{u}_i^l), & (\tilde{u}_{i+1}^l - \tilde{u}_i^l)(u_{i+1}^l - u_i^l) < 0, \\ \frac{1}{2}(\tilde{u}_{i-1}^l + \tilde{u}_i^l), & (\tilde{u}_{i-1}^l - \tilde{u}_i^l)(u_{i-1}^l - u_i^l) < 0, \\ \tilde{u}_i^l, & \text{else} \end{cases} \quad (5.27)$$

with the initial conditions (5.21) and boundary conditions as in the previous Time-Discrete Problem.

Note that the case distinction on the right-hand side of (5.27) is sound since the case (ii) of Proposition 5.5 cannot occur simultaneously on both sides of the same pixel. Further, it is clear from Proposition 5.5 that the Modified Time-Discrete Problem is identical with the Time-Discrete Problem for $\tau < 1/2$.

For the Modified Time-Discrete Problem, the well-posedness statements of Theorem 5.4 hold for all $\tau \leq 1$. Since the proof contains only slight modifications compared to the previous one, we do not repeat it here but state only the suitably modified version of Proposition 5.5. The main modification is that *extremal* linear pixels can now arise during the evolution.

Proposition 5.6 (Modified Time-Discrete Solution) Let u_i^l be the value of pixel i in time step l of the solution of our Modified Time-Discrete Problem with time step size $\tau \leq 1$. Then the following hold for all $l = 0, 1, 2, \dots$:

- (i) If u_1^l is a local maximum of u^l , then u_1^{l+1} is a local maximum of u^{l+1} .
- (ii) If u_1^l is a concave pixel neighbouring to a convex pixel $u_0^l > u_1^l$, then u_1^{l+1} is again concave and has a convex neighbour pixel $u_0^{l+1} \geq u_1^{l+1}$.

- (iii) If the segment (u_1^l, \dots, u_m^l) is strictly decreasing and concave in all pixels, and u_1^l is either a local maximum of u^l or neighbours to a convex pixel $u_0^l > u_1^l$, then the segment $(u_2^{l+1}, \dots, u_m^{l+1})$ is strictly decreasing, and $u_1^{l+1} \geq u_2^{l+1}$.
- (iv) Under the same assumptions as in (iii), the segment $(u_2^{l+1}, \dots, u_m^{l+1})$ is strictly concave in all pixels. Pixel u_1^{l+1} is strictly concave except if u_1^l is a local maximum, and $\tau = 1$.
- (v) If $2u_i^l = u_{i+1}^l + u_{i-1}^l$ holds for no pixel i for which $u_{i-1}^l = u_i^l = u_{i+1}^l$ does not hold, then $2u_i^{l+1} = u_{i+1}^{l+1} + u_{i-1}^{l+1}$ also holds for no pixel i for which $u_{i-1}^{l+1} = u_i^{l+1} = u_{i+1}^{l+1}$ does not hold.
- (vi) Under the assumptions of (iii), all pixels in the range $i \in \{1, \dots, m\}$ have the same limit $\lim_{l \rightarrow \infty} u_i^l = C$ with C as in (5.22).

Analogous statements hold for local minima, for increasing concave and for convex signal segments.

The special case $\tau = 1$ deserves a closer consideration. Straightforward calculations reveal that a pixel neighbouring to a local maximum attains the same value as the maximum in the next time step. Moreover, a pair of a convex pixel followed by a concave one in a decreasing segment aligns to equal values within one time step, turning both pixels into discrete local extrema. These facts give rise to the following corollary.

Corollary 5.7 Consider the Modified Time-Discrete Problem with $\tau = 1$. If at time step $k = l$ the segment $(u_1^l, u_2^l, \dots, u_m^l)$ has the properties required in Prop. 5.6, (iii), then each pixel u_k of this segment becomes constant after not more than k time steps.

In the case of finite-length signals, or infinite signals in which there exists a finite upper bound to the length of monotonic concave or convex segments, this corollary implies that the steady state is reached in finite time.

Since the modified time-discrete filter propagates grey-values in x direction one pixel per time step, it turns out to reflect particularly well the behaviour of dilations and erosions on a continuous domain. In the light of our remark following the proof of Theorem 5.1, it can be said that the approximation error introduced by spatial discretisation has been compensated exactly by the time discretisation.

A further view on the modified scheme is that it can be related to the locally analytic schemes from Chapter 4. Our modified time-discrete shock filtering scheme can be understood as a locally analytic scheme built on the analytical solutions (for $0 \leq \tau \leq 1$) of *space-continuous* dilation and erosion filters where the signal is linearly interpolated between subsequent pixels.

5.1.3 Equivalence to Local Mode Filtering

Now that we have derived well-posedness properties for semidiscrete and fully discrete shock filters, let us also establish an equivalence result between shock filters and a class of discrete filters based on local signal statistics, namely so-called mode filters.

Discrete filters exploiting local signal statistics evaluate signal values from a sliding window neighbourhood to determine a new value for each pixel. Commonly used representatives of this class are box-average filters, median filters and generally M-smoothers, but also discrete dilation and erosion.

One statistical parameter of the local greyvalue distribution that is not used in one of the aforementioned filters is the *mode*. The mode of a continuous distribution is defined as its most probable value. Analogous as above, determining the mode of the grey-values within a sliding window or structuring element constitutes a local mode filter for images [71, 98, 194].

However, applying this procedure to spatially discretised signals faces a problem because the distribution is now given only by finitely many values. Defining the mode simply as the most frequent value is not helpful since in generic cases there are no duplicates among the values.

Instead, we combine a polynomial approximation with local histogram properties to find the mode value within a sliding window containing three pixels.

Assume we have a one-dimensional discrete signal $(\dots, u_0^l, u_1^l, u_2^l, \dots)$. By our sliding window we select three subsequent values $u_{i-1}^l, u_i^l, u_{i+1}^l$. The value at pixel i of the signal filtered by our local mode filter should be the mode value of $u_{i-1}^l, u_i^l, u_{i+1}^l$.

To begin with, we interpolate by a quadratic polynomial through the three points

$$(i-1, u_{i-1}^l), \quad (i, u_i^l), \quad (i+1, u_{i+1}^l). \quad (5.28)$$

Translating, for simplicity, spatial coordinates by $-i$, we therefore want the quadratic polynomial

$$p(z) = az^2 + bz + c \quad (5.29)$$

to satisfy the conditions

$$p(-1) = u_{i-1}^l, \quad p(0) = u_i^l, \quad p(1) = u_{i+1}^l. \quad (5.30)$$

This gives the system of three equations

$$\begin{aligned} a - b + c &= u_{i-1}^l, \\ c &= u_i^l, \\ a + b + c &= u_{i+1}^l, \end{aligned} \quad (5.31)$$

with the solution

$$a = \frac{u_{i-1}^l - 2u_i^l + u_{i+1}^l}{2}, \quad b = \frac{u_{i+1}^l - u_{i-1}^l}{2}, \quad c = u_i^l. \quad (5.32)$$

Having determined $p(z)$, we are now interested in the location of the mode of its values. First, if $a = 0$, p is a linear polynomial whose values are uniformly distributed. In this case, we have our local mode filter not change the value of pixel i .

If $a \neq 0$, the density of the distribution of values of p attains its maximum at the (uniquely determined) stationary value of p . The extremum of $p(z)$ is located at

$$-\frac{b}{2a} = \frac{u_{i-1}^l - u_{i+1}^l}{u_{i-1}^l - 2u_i^l + u_{i+1}^l} =: e. \quad (5.33)$$

However, whenever $e \notin \{-1, 0, +1\}$, the extremal value $p(e)$ will lie outside the interval $I_i := [\min(u_{i-1}^l, u_i^l, u_{i+1}^l), \max(u_{i-1}^l, u_i^l, u_{i+1}^l)]$; choosing $p(e)$ as the value of the mode filter therefore results in over- and undershoots.

This leads us to stabilise our filtering procedure by projecting $p(e)$ to the interval I_i , i.e., choosing as the new value of pixel i the value from the interval I_i which is closest to $p(e)$. We will call this procedure *stabilised discrete local mode filtering*. Clearly, for p not linear this filter will always return as its value one of the end points of I_i , namely $\min(u_{i-1}^l, u_i^l, u_{i+1}^l)$ if $a < 0$, or $\max(u_{i-1}^l, u_i^l, u_{i+1}^l)$ if $a > 0$.

We have therefore arrived at the following equivalence result.

Proposition 5.8 *Stabilised discrete local mode filtering of a 1-D discrete signal $(\dots, f_0, f_1, f_2, \dots)$ obeys the equation*

$$u_i = \begin{cases} \min(u_{i-1}^l, u_i^l, u_{i+1}^l), & u_{i-1}^l - 2u_i^l + u_{i+1}^l < 0, \\ \max(u_{i-1}^l, u_i^l, u_{i+1}^l), & u_{i-1}^l - 2u_i^l + u_{i+1}^l > 0, \\ u_i^l, & u_{i-1}^l - 2u_i^l + u_{i+1}^l = 0. \end{cases} \quad (5.34)$$

Consequently, it is equivalent to one step of time-discrete shock filtering as described in 5.1.2 with time step size $\tau = 1$.

Since the time step size $\tau = 1$ exceeds the limit given in Theorem 5.4, the well-posedness properties do not transfer fully to stabilised discrete local mode filtering. However, a modification analog to the modified scheme of 5.1.2 could also be applied to local mode filtering to obtain a well-posed process.

5.2 Stabilised Inverse Linear Diffusion

In this section, we are concerned with the processes governed by time-dependent PDEs of the form

$$\frac{\partial}{\partial t} u(x, t) = \begin{cases} -c \frac{\partial^2}{\partial x^2} u(x, t), & \frac{\partial}{\partial x} u(x, t) \neq 0 \\ 0, & \frac{\partial}{\partial x} u(x, t) = 0 \end{cases} \quad (5.35)$$

where u is a scalar-valued signal, $x \in \mathbb{R}$, $t > 0$, and $c > 0$ is a constant anti-diffusion coefficient. We will call these processes *stabilised inverse linear diffusion (SILD)*.

Their appearance in an image processing context goes back to [156]. Since then, they have been used as building blocks of image filters, see e.g. [95] and the references therein. Outside image processing, models of the form (5.35) have been used as sharpening filters within flux-corrected transport (FCT) schemes in the field of computational fluid dynamics [33, 127].

Naturally, our considerations also link once more to the *stabilised inverse diffusion equations (SIDEs)* introduced in [164, 165] where, however, emphasis is laid on non-linear inverse diffusion processes.

In this section, we are particularly interested in the analysis of a numerical phenomenon which can spoil the results of SILD filtering, namely the so-called *staircasing phenomenon* [230]: when applied to a strictly monotone signal of finite length, a numerically realised SILD process typically generates a stepfunction-type solution reminding of a staircase instead of a new strictly monotone signal with sharper gradient. Since [230] where staircasing was described for Perona–Malik diffusion [161], it has been observed in nonlinear diffusion processes involving backward diffusion, see e.g. [210] and references therein. For staircasing in the context of total variation regularisation, we refer to [153, 172]. To overcome or at least reduce staircasing, special adaptive diffusion processes have been designed, see e.g. [32, 46, 65, 116, 131, 210].

However, the staircasing phenomenon by itself has not been analysed mathematically in much detail up to now. It is useful to note again in this context that realistic initial signals consist of monotone segments of finite length, which have a local maximum and a local minimum as end-points. Simple infinite models of staircasing which do not take into account the corresponding effects, like e.g. the lever model mentioned in [127], do not give a full account of the effects that are in fact observed in numerical computations.

Recently, staircasing in fully discrete schemes for PDEs of type (5.35) has been investigated in [36]. It has been proven that it necessarily occurs in these schemes. In the context of fully discrete SILD processes, the question arises if staircasing could be avoided either by some specific time stepping scheme, or by using very small time steps in the time integration. In order to better understand the nature of staircasing as

well as for the development of algorithms, it is of interest whether staircasing events follow some mechanism that can be determined in advance. A closely related question is staircasing is a numerically stable phenomenon – in other words: Can small perturbations due to low-level noise or numerical errors lead to significant changes in the result of a numerical approximation of (5.35)?

In order to clarify the meaning of these open points, let us briefly discuss fully discrete approximations of (5.35). To this end, we set $u_i^n \approx u(i, n\tau)$ using a space-time grid with corresponding, uniform grid parameters $h := 1$ and τ , and denote by $g_{i\pm 1/2}$ consistent numerical fluxes at the boundaries between the cells i and $i\pm 1$, respectively. Note that this flux discretisation at inter-pixel locations corresponds to the procedure in 4.1. Then, we obtain as a consistent and conservative approximation of (5.35) the process

$$u_i^{n+1} = u_i^n - \tau(g_{i+1/2} - g_{i-1/2}). \quad (5.36)$$

which is also in analogy to (the time discretisation of) our scheme in 4.1, with the divergence form discretisation guaranteeing conservativity.

Stabilisation in this setting takes place via an appropriate choice of the fluxes. To this end, we employ the minmod-function

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } a \cdot b > 0 \text{ and } |a| \leq |b|, \\ b & \text{if } a \cdot b > 0 \text{ and } |b| \leq |a|, \\ 0 & \text{else,} \end{cases} \quad (5.37)$$

which can easily be extended to more than two arguments if necessary, compare e.g. [130], by choosing the argument with minimal modulus if the arguments are of the same sign, and zero else. The most basic useful scheme for approximating (5.35) then incorporates the minmod function by

$$g_{i+1/2} = \text{minmod}\left(\frac{1}{\tau}(u_{i+2}^n - u_{i+1}^n), c(u_{i+1}^n - u_i^n), \frac{1}{\tau}(u_i^n - u_{i-1}^n)\right). \quad (5.38)$$

Let us stress that the natural discretisation of fluxes in (5.35) is given by means of the middle argument of (5.38),

$$c(u_{i+1}^n - u_i^n), \quad (5.39)$$

and its counterpart in $g_{i-1/2}$. The other ingredients of $g_{i\pm 1/2}$ act as stabilisers, which is easily recognised by taking into account the multiplication by τ from (5.36).

Let us now consider inner points of a strictly monotone data set $u^0 := \{u_k^0, \dots, u_l^0\}$. We assume that the extrema u_k^0 and u_l^0 stay fixed, and we point out that the events

$$u_{k+1}^{n_1} = u_k^0 \quad \text{or} \quad u_{l-1}^{n_2} = u_l^0 \quad \text{for any } n_1, n_2 \geq 1 \quad (5.40)$$

do not constitute staircasing as staircasing refers to steps aka constant valued data tuples arising away from extrema. Especially, one may consider to choose the time

step size τ small enough such that the method reduces for many time steps at inner points of u^0 to

$$u_i^{n+1} = u_i^n - c\tau \left((u_{i+1}^n - u_i^n) - (u_i^n - u_{i-1}^n) \right), \quad i = k+2, \dots, l-2, \quad (5.41)$$

i.e., in such cases the numerical fluxes reduce to the middle arguments, (5.39), such that no stabilisation takes effect at these points. Heuristically, one then expects that no staircasing occurs in strictly monotone data regimes: numerical fluxes $g_{i\pm 1/2}$ as in (5.39) always introduce nonzero updates for $i = k+2, \dots, l-2$, whenever data are not distributed exactly along a linear segment where $u_{i+1}^n - u_i^n = u_i^n - u_{i-1}^n$. Also the other mentioned aspects of interest are close to this line of argumentation: since one cannot avoid to employ the minmod stabilisation, as is shown in [36], it is natural to assume that the number and position of staircasing artifacts depends on how often the minmod stabilisation takes effect. This frequency in turn could be influenced, e.g., by manipulating the time step size.

In the following, however, it will become evident that the stated expectations are not based on solid ground.

- Staircasing arises already in semidiscrete approximations of (5.35) and is merely bequeathed to fully discrete methods which approximate the semidiscrete process.
- Concerning stability, we show that semidiscrete SILD processes involve bifurcations, such that the stability of numerical results under small data perturbations cannot be guaranteed for all data configurations.
- Furthermore, we show that the choice of a time stepping method is also problematic: a naive proceeding can easily violate important invariance properties of the semidiscrete formulation of (5.35), namely that the total variation as well as the number of extrema of a given signal do not increase during time evolution.

5.2.1 Semidiscrete Analysis

We will now formulate and analyse a dynamical system that describes a spatially discretised, time-continuous version of SILD. We use the spatial discretisations of flux and diffusion that have already been described. For simplicity we assume from now on $c = 1$.

5.2.1.1 The Dynamical System

We consider real-valued, time-dependent signals

$$u = u(t) = (\dots, u_0(t), u_1(t), u_2(t), \dots) \quad (5.42)$$

of compact support. The latter restriction can be relaxed; it is sufficient to ensure that signals are bounded and do not contain strictly monotone segments of infinite length. Note that the time-continuous functions $u_i(t) \approx u(i, t)$ defined at discrete points in space is distinguished from fully discrete data $u_i^n \approx u(i, n\tau)$ by the absence of the upper, time-step index. We assume that spatial and temporal discretisations are uniform with mesh sizes $1, \tau$, respectively.

Proceeding as in the fully discrete setting, see (5.36)–(5.38), we obtain for a conservative evolution on a signal (5.42) the following dynamical system of ordinary differential equations

$$\dot{u}_i = g_{i-1/2} - g_{i+1/2}. \quad (5.43)$$

Here, $g_{i+1/2} = g_{i+1/2}(u, t)$ is the flux between adjacent pixels i and $i + 1$, which we assume to fulfil the following conditions.

Translational invariance: $g_{i+1/2}(u, t) = g_{1/2}(S_{-i}(u), t)$, where $S_{-i}(u)$ denotes the signal u shifted by $-i$ pixels, $(S_{-i}(u))_j = u_{j+i}$,

Time invariance: $g_{1/2}(u, t) = g_{1/2}(u)$. This assumption means that the system (5.43) is *autonomous*.

Inverse diffusion without stabilisation can be realised by

$$g_{1/2} = -u_0 + u_1 \quad (5.44)$$

leading to

$$\dot{u}_i = -u_{i-1} + 2u_i - u_{i+1}. \quad (5.45)$$

A *stabilisation* is introduced in order to ensure that local extrema become invariant values, compare also the discussion in [36] for the fully discrete case. This means, that (5.44) is used only if neither u_0 nor u_1 is an extremum, otherwise we set $g_{1/2} = 0$.

The stabilised version of (5.45) thus reads as the system

$$\dot{u}_i = \begin{cases} (-u_{i-1} + 2u_i - u_{i+1}), & (u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}) \text{ strictly} \\ & \text{monotone} \\ (u_i - u_{i+1}), & (u_{i-1}, u_i, u_{i+1}, u_{i+2}) \text{ strictly monotone} \\ & \text{and } u_{i-1} \text{ local extremum} \\ (-u_{i-1} + u_i), & (u_{i-2}, u_{i-1}, u_i, u_{i+1}) \text{ strictly monotone} \\ & \text{and } u_{i+1} \text{ local extremum} \\ 0 & \text{else.} \end{cases} \quad (5.46)$$

Our notion of a discrete extremum is the same as in 5.1: any pixel u_i for which the sequence (u_{i-1}, u_i, u_{i+1}) is not strictly monotone is considered as local extremum (compare remark on page 173).

The equations (5.46) comprise a dynamical system with discontinuous right-hand side. Let us therefore first specify our concept of solution, analogous to the procedure in 4.1.1 and 5.1.

Given an initial signal $(\dots, f_0, f_1, f_2, \dots)$, we will say that a time-dependent signal $(\dots, u_0(t), u_1(t), u_2(t), \dots)$ is a solution of the initial-value problem consisting of the differential equations (5.46) and the initial conditions

$$u_i(0) = f_i, \quad i = \dots, 0, 1, 2, \dots, \quad (5.47)$$

if the following conditions are satisfied:

- (I) each u_i is a continuous, piecewise differentiable function of t ,
- (II) each u_i satisfies (5.46) for all t for which $\dot{u}_i(t)$ exists,
- (III) for $t = 0$, the right-sided derivative $\dot{u}_i^+(0)$ equals the right-hand side of (5.46).

In the following, we will refer to (5.46) *with this understanding of solution* as *semidiscrete stabilised inverse linear diffusion*.

5.2.1.2 Analytical Solution

We start by noting the following facts.

Lemma 5.9 *Let $(\dots, u_0(t), u_1(t), u_2(t), \dots)$ be a solution of (5.46)–(5.47) in the sense of (I)–(III).*

Then the following hold:

- (i) *If u_i is a local extremum at a time $t = t_0$, its neighbours u_{i-1} and u_{i+1} can not move away from u_i at $t = t_0$.*
- (ii) *If u_i is a local extremum at a time t_0 , it remains a local extremum for all $t > t_0$.*
- (iii) *If $u_i = u_{i+1}$ at a time t_0 , then the same equality holds for all $t > t_0$.*

Remark. The Lemma implies particularly the preservation of monotonicity, thereby guaranteeing that the process is *total variation preserving (TVP)*, compare [130] for this notion.

Proof. We prove that, as long as u_i is a local extremum, its neighbour u_{i+1} can move only towards u_i . Indeed, in case u_i is a local extremum we have, see (5.46),

$$\dot{u}_{i+1} = \begin{cases} (u_{i+1} - u_{i+2}) & \text{if neither } u_{i+1} \text{ nor } u_{i+2} \text{ is a local extremum,} \\ 0 & \text{else.} \end{cases} \quad (5.48)$$

If thus \dot{u}_{i+1} is to be non-zero, u_{i+1} cannot be an extremum, and $u_{i+1} - u_i$ and $u_{i+2} - u_{i+1}$ have the same sign. Consequently,

$$\operatorname{sgn}(\dot{u}_{i+1}) = \operatorname{sgn}(u_i - u_{i+1}) \quad (5.49)$$

holds at any time t in case u_i is an extremum. Note, that the left neighbour u_{i-1} of u_i can be treated in an analogous fashion. This proves statement (i).

Next, we prove that pixels, once they have attained the same value, cannot split up again to attain different values. Assume there were two neighbouring pixels u_i and u_{i+1} which are equal at time t_0 and unequal at time $t_1 > t_0$, without loss of generality we set $u_{i+1}(t_1) > u_i(t_1)$. Furthermore, we assume that the interval (t_0, t_1) is chosen such that the signs of differences $u_{i+2} - u_{i+1}$ and $u_{i-1} - u_i$ do not change within the interval, and such that u_i and u_{i+1} are differentiable throughout the interval (t_0, t_1) . Note that this can always be ensured by splitting the interval if necessary. According to the mean-value theorem of differential calculus, there exists a $\vartheta \in (t_0, t_1)$ such that

$$(t_1 - t_0)(\dot{u}_{i+1}(\vartheta) - \dot{u}_i(\vartheta)) = (u_{i+1}(t_1) - u_i(t_1)) - (u_{i+1}(t_0) - u_i(t_0)), \quad (5.50)$$

which entails

$$\dot{u}_{i+1}(\vartheta) - \dot{u}_i(\vartheta) > 0. \quad (5.51)$$

To prove assertion (ii) of the Lemma we derive a contradiction to (5.51).

If u_i is a local minimum at $t = \vartheta$ (thus, throughout (t_0, t_1)), we have that u_{i+1} is not an extremum, and it follows that

$$\dot{u}_{i+1}(\vartheta) - \dot{u}_i(\vartheta) = u_{i+1} - u_{i+2} < 0. \quad (5.52)$$

An analogous argument holds if u_{i+1} is a local maximum and u_i not an extremum.

Finally, if neither u_i nor u_{i+1} is an extremum at $t = \vartheta$, we have that the differences $u_{i-1} - u_i$, $u_i - u_{i+1}$, $u_{i+1} - u_{i+2}$ are all negative in (t_0, t_1) , and at least one of $u_{i-1} - u_i$, $u_{i+1} - u_{i+2}$ is negative for $t = t_0$. By choosing the interval (t_0, t_1) small enough, we can achieve that $2(u_i - u_{i+1}) > u_{i-1} - u_i + u_{i+1} - u_{i+2}$ throughout (t_0, t_1) , from which it follows that

$$\dot{u}_{i+1}(\vartheta) - \dot{u}_i(\vartheta) = (u_{i+1} - u_{i+1}) - 2(u_i - u_{i+1}) + (u_{i-1} - u_i) < 0. \quad (5.53)$$

To sum up, we have obtained contradictions to (5.51) in all possible cases, which completes the proof of statement (ii).

The third assertion follows from the fact that for a local extremum to lose its extremality, it would have to be “passed” by one of its neighbours, which would therefore have to be equal to the extremum at some time (remember u is continuous with respect

to t). According to the second statement, the two pixels would irreversibly merge in this case. \square

It is therefore sufficient to consider the evolution of signal segments of finite length whose first and last pixels are local extrema, and which are strictly monotone. Without loss of generality, we consider a decreasing segment $f_0 > f_1 > \dots > f_n > f_{n+1}$ where f_0 is a local maximum and f_{n+1} a local minimum. Thus, we have the evolution equations

$$\begin{aligned} \dot{u}_0 &= 0, \\ \dot{u}_1 &= u_1 - u_2, \\ \dot{u}_i &= -u_{i-1} + 2u_i - u_{i+1}, \quad 2 \leq i \leq n-1, \\ \dot{u}_n &= -u_{n-1} + u_n, \\ \dot{u}_{n+1} &= 0, \end{aligned} \tag{5.54}$$

which hold throughout any time interval $(0, T)$ in which $u_0 > u_1 > \dots > u_n > u_{n+1}$ stays true.

The system (5.54) is a system of linear ODEs which can be solved analytically. Leaving aside u_0 and u_{n+1} , we can rewrite the system for $\mathbf{u} := (u_1, \dots, u_n)^T$ as

$$\dot{\mathbf{u}} = A\mathbf{u} \tag{5.55}$$

with the $n \times n$ matrix

$$A = \begin{pmatrix} 1 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}. \tag{5.56}$$

The matrix A is positive semidefinite, since by Gershgorin's Theorem [93] all its eigenvalues are nonnegative. Moreover, A has rank $n-1$ since it contains a triangular $(n-1) \times (n-1)$ submatrix without zeros on its diagonal. We therefore simplify the system by eliminating the zero eigenvalue and corresponding eigenvector.

For $v_0 := \frac{1}{\sqrt{n}} \sum_{i=1}^n u_i$ we have $\dot{v}_0 = 0$, implying $v_0(t) = v_0(0)$ for all t . In fact, $v_0 = a_0^T \mathbf{u}$ where $a_0 = \frac{1}{\sqrt{n}}(1, \dots, 1)^T$ is the eigenvector with eigenvalue zero of A .

Let us now set $v_i := u_i - u_{i+1}$, $i = 1, \dots, n-1$, and $\mathbf{v} := (v_1, \dots, v_{n-1})^T$, i.e.,

$$\mathbf{v} := D\mathbf{u}, \tag{5.57}$$

with the $(n-1) \times n$ matrix

$$D = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix}. \quad (5.58)$$

Introducing additionally the $(n-1) \times (n-1)$ matrix

$$B = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix}, \quad (5.59)$$

one easily sees that

$$D^T D = A, \quad D D^T = B. \quad (5.60)$$

Thus, by (5.55) and (5.60), we obtain

$$D \dot{u} = D A u = B D u, \quad (5.61)$$

i.e., a new linear dynamical system for v :

$$\dot{v} = B v. \quad (5.62)$$

The analytical solution of (5.62) is given by

$$v(t) = e^{Bt} v(0) \quad (5.63)$$

which we will make more explicit using the eigendecomposition of the symmetric matrix B .

Lemma 5.10 *Let $\delta_k := \pi k/n$. Then*

$$b_k := \sqrt{\frac{2}{n}} (\sin(\delta_k), \sin(2\delta_k), \dots, \sin((n-1)\delta_k))^T \quad (5.64)$$

for $k = 1, \dots, n-1$ are normalised eigenvectors of B , with corresponding eigenvalues

$$\lambda_k = 2(1 - \cos(\delta_k)). \quad (5.65)$$

Remark. The matrix B represents a discrete Laplacian with zero boundary conditions. Consequently, its eigenvectors are discretised harmonic functions, namely the basis of a discrete sine transform.

Proof. By direct calculation one checks that each b_k is of unit length and satisfies

$$B b_k = 2(1 - \cos(\delta_k)) b_k. \quad (5.66)$$

□

Via

$$v(t) = \sum_{k=1}^{n-1} \langle b_k, v(0) \rangle b_k e^{\lambda_k t} \quad (5.67)$$

we can rewrite (5.63) to obtain directly the following statement.

Proposition 5.11 For $t \in [0, T]$ the solution of (5.62) is given by

$$v_i(t) = \frac{2}{n} \sum_{j=1}^{n-1} \left(\sum_{k=1}^{n-1} \sin \frac{\pi i k}{n} \sin \frac{\pi j k}{n} e^{2(1 - \cos \frac{\pi k}{n})t} \right) v_j(0). \quad (5.68)$$

The analytical solution of (5.55) is then computed by backsubstituting u for v and it is given within the following corollary.

Corollary 5.12 The solution of (5.55) for $t \in [0, T]$ is given by

$$u_i(t) = \frac{1}{n} \left(\sum_{j=1}^n u_j(0) - \sum_{j=1}^{i-1} j v_j(t) + \sum_{j=i}^{n-1} (n-j) v_j(t) \right), \quad (5.69)$$

where $v_j(t)$ are given by (5.68), and thus by

$$u_i(t) = \sum_{j=1}^n \left(\frac{1}{n} + \frac{4}{n^2} \sum_{k=1}^{n-1} \cos \frac{\pi(2j-1)k}{2n} \sin \frac{\pi k}{2n} e^{2(1 - \cos \frac{\pi k}{n})t} \cdot \left(\sum_{l=1}^{n-1} l \sin \frac{\pi l k}{n} + n \sum_{l=i}^{n-1} \sin \frac{\pi l k}{n} \right) \right) u_j(0). \quad (5.70)$$

In (5.69) and (5.70), sums with upper limit below lower limit are to be read as zero.

Remark. The evolution (5.55) (or also (5.70)) can also be read as non-stabilised inverse linear diffusion on a finite signal (u_1, \dots, u_n) with reflecting, i.e. zero-flux, boundary conditions. That is to say, in time intervals between pixel merging events the strictly monotone segments of semidiscrete stabilised inverse linear diffusion follow an ordinary inverse linear diffusion dynamics; at merging events, just the segmentation changes.

5.2.1.3 Staircasing in Segments

We continue considering a strictly decreasing signal segment enclosed between two local extrema, and we want to determine under which conditions staircasing occurs. We start with the following observation.

Lemma 5.13 *Let a strictly decreasing segment (f_0, \dots, f_{n+1}) with local extrema f_0 and f_{n+1} be given, $n \geq 2$, and let (u_0, \dots, u_{n+1}) evolve according to (5.46) with initial condition $u(0) = f$. Then the dynamics of (u_1, \dots, u_n) follows (5.55) until one of the following events happens:*

- (a) *One of the pixels u_1, u_n merges with its extremal neighbour pixel u_0, u_{n+1} , respectively.*
- (b) *Two neighbouring pixels u_i, u_{i+1} ($1 \leq i \leq n-1$) become equal.*

Either (a) or (b) occurs for a finite $t = T$.

Case (b) describes a staircasing event, entailing a transition to two smaller segments, while in case (a) a transition to a smaller segment is made without staircasing. Even in the latter case, a later staircasing event involving the same pixels is still possible but then governed by the dynamics of the new segments.

Proof. It is clear that the dynamics (5.55) is terminated as soon as (a) or (b) occurs. It remains to show that whatever initial values are given, this happens at finite evolution time $t = T$.

From Lemma 5.9 it follows that for $n \geq 2$ both u_1 and u_n evolve in direction of their neighbouring extrema as long as no staircasing involving these pixels has occurred, i.e., as long as $u_1 > u_2$ and $u_{n-1} > u_n$ hold. Since (5.68) contains only exponential summands with positive exponents, it is clear that the velocity of both pixels cannot go asymptotically to zero; thus, whatever values u_0 and u_{n+1} might have (which don't influence the dynamics of u_1, \dots, u_n directly), either u_1 or u_n will merge with its neighbouring extremum in finite time. \square

However, by choosing f_0 large enough and f_{n+1} small enough, the end-of-segment merging events can be pushed to arbitrarily large values of t . This leads us to ask: *For which values of u_1, \dots, u_n is the dynamics (5.55) guaranteed to be terminated by an end-of-segment merging event of type (a), independent on f_0 and f_{n+1} ?*

To answer this question, we consider the dynamics (5.55) just as ordinary semidiscrete inverse diffusion with zero-flux boundary conditions, and focus on the differences v_1, \dots, v_{n-1} . These differences are positive at $t = 0$, and staircasing events are

indicated by at least one of these differences reaching zero. We can then prove the following result.

Proposition 5.14 *Given a strictly decreasing signal (f_1, \dots, f_n) , the dynamics (5.55) with initial condition $u(0) = f$ preserves the strict monotonicity $u_1 > \dots > u_n$ for all $t > 0$ if and only if the differences $v_1(0) = f_1 - f_2, \dots, v_{n-1}(0) = f_{n-1} - f_n$ are given by some multiple μb_1 of the eigenvector b_1 with $\mu > 0$.*

The proof relies on two important properties of the eigendecomposition of B which can be directly read off the formulae (5.64), (5.65).

Lemma 5.15 *For the eigenvectors and eigenvalues of B given by (5.64), (5.65), the following properties hold:*

1. *The eigenvalues are ordered by size, $\lambda_1 < \lambda_2 < \dots < \lambda_{n-1}$.*
2. *Exactly one eigenvector, namely b_1 which corresponds to the smallest eigenvalue, has only positive components. Each of the eigenvectors b_2, \dots, b_{n-1} has at least one negative component.*

Proof of Proposition 5.14. Since $v(0)$ has positive components, $\langle v(0), b_k \rangle \neq 0$ holds for some k . Let k be the largest index with this property. Considering (5.67) for $t \rightarrow \infty$, we have that

$$\lim_{t \rightarrow \infty} \frac{v(t)}{e^{2\lambda_k t}} = \langle v(0), b_k \rangle b_k. \quad (5.71)$$

Assuming that $v_i > 0$ for all i and all t , the limits on the left hand side of (5.71) must be nonnegative which can only be the case if all components of b_k are nonnegative, or if all are nonpositive. According to Lemma 5.15 this implies $k = 1$. \square

Initial values f which do not satisfy the condition from Proposition 5.14 can be classified depending on which neighbouring values in the signal will merge first. We describe this classification qualitatively in terms of the difference variables v .

To this end, we note that (5.63) can be evaluated for negative t as well as for positive t since the linear system (5.62) is *reversible*. This implies that all initial values v° which lead to a given state v^* at later times can be obtained, vice versa, by (5.63) if $v(0) = v^*$ is used as an initial condition and going backwards in time.

Since we seek to investigate which v_i vanishes first during evolution, we want to know *where* trajectories leave the sector $(\mathbb{R}_0^+)^{n-1}$. The boundary of this sector is made up by $n - 1$ *facets*, each of them characterised by one of the variables v_i , $i \in \{1, \dots, n - 1\}$ attaining zero value, see Figure 5.2.

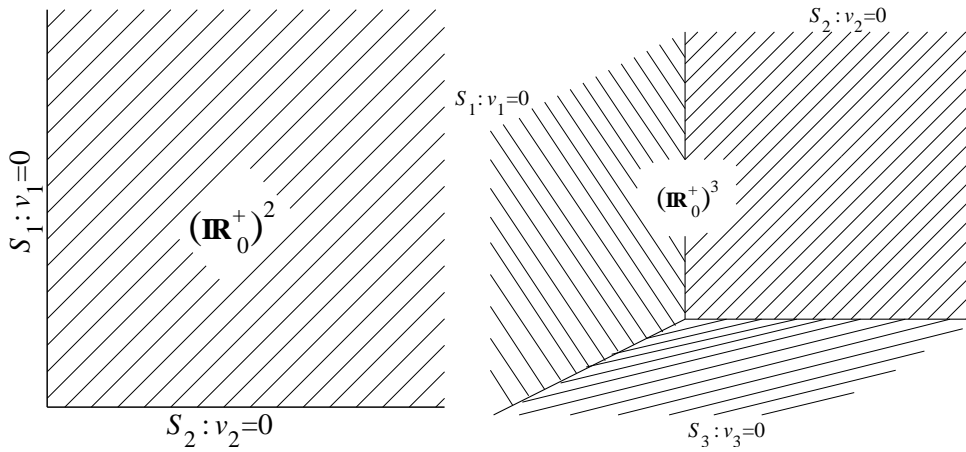


Figure 5.2: Facets of the boundary of $(\mathbb{R}_0^+)^n$. Left: (a) $n = 3$. Right: (b) $n = 4$.

Let the facet consisting of all these points $(v_0 > 0, \dots, v_i = 0, \dots, v_{n-1} > 0)^T$ be denoted by S_i . Each facet is simply connected. Denoting by $T_{t<0}(v)$ the trajectory of a point $v \in \mathbb{R}^{n-1}$ propagating under (5.63) backwards in time, we see that the set of initial conditions for which v_i is the first variable to vanish during evolution is exactly

$$T_{t<0}(S_i) := \bigcup_{v \in S_i} T_{t<0}(v). \quad (5.72)$$

Note that solutions of (5.63) are continuous in t . Moreover, they depend continuously on initial conditions, and because of the reversibility of the system, trajectories are either identical or disjoint. The union $T_{t<0}(S_i)$ of negative trajectories starting on a single facet S_i is therefore a simply connected $(n-1)$ -dimensional point set in \mathbb{R}^{n-1} , whose boundary consists of the facet S_i itself and those trajectories starting on the boundary of S_i ,

$$\partial T_{t<0}(S_i) = S_i \cup T_{t<0}(\partial S_i), \quad (5.73)$$

compare Figure 5.3.

Different sets $T_{t<0}(S_i), T_{t<0}(S_j)$ are therefore separated by hypersurfaces $T_{t<0}(S_{ij}), S_{ij} := \partial S_i \cap \partial S_j$ (except for $n = 3$ where the separating line cannot be obtained from $S_{12} = \{0\}$ in this way, see instead the discussion below for this case). The topology of the resulting partition of $(\mathbb{R}_0^+)^{n-1}$ is therefore equivalent to the topology of the $(n-1)$ -dimensional surface of a n -dimensional (hyper)cube corner with one facet removed.

We notice further that if $v \in T_{t<0}(S_i)$, then the linearity of (5.63) implies $\alpha v \in T_{t<0}(S_i)$ for any $\alpha > 0$. This ensures that any hyperplane $H = H_{n,C}$ defined by $\langle n, v \rangle = C$ (where $\langle \cdot, \cdot \rangle$ denotes Euclidean scalar product, $n \in (\mathbb{R}_0^+)^{n-1}$ and $C > 0$ are fixed) is transversal to all trajectories under consideration. Thus, the separation of $(\mathbb{R}_0^+)^{n-1}$ induces by restriction a separation of $(\mathbb{R}_0^+)^{n-1} \cap H$ whose topology equals

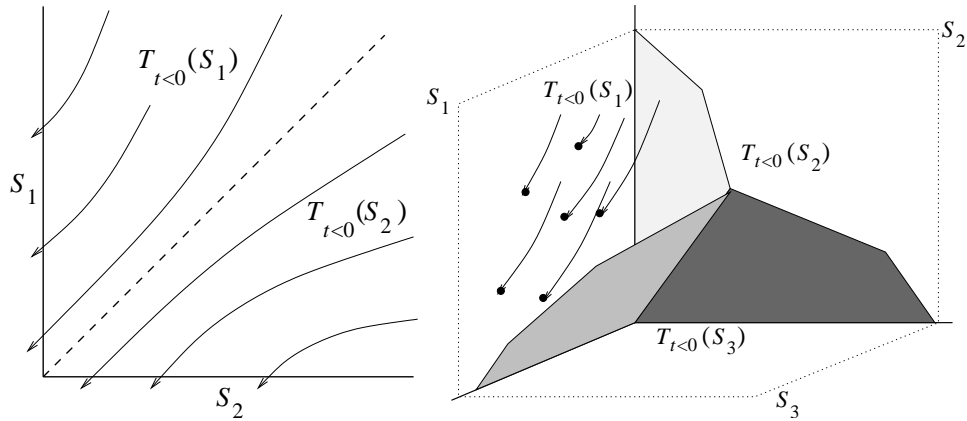


Figure 5.3: Partition of $(\mathbb{R}_0^+)^n$ into regions $T_{t<0}(S_i)$. Left: (a) $n = 3$. Right: (b) $n = 4$.

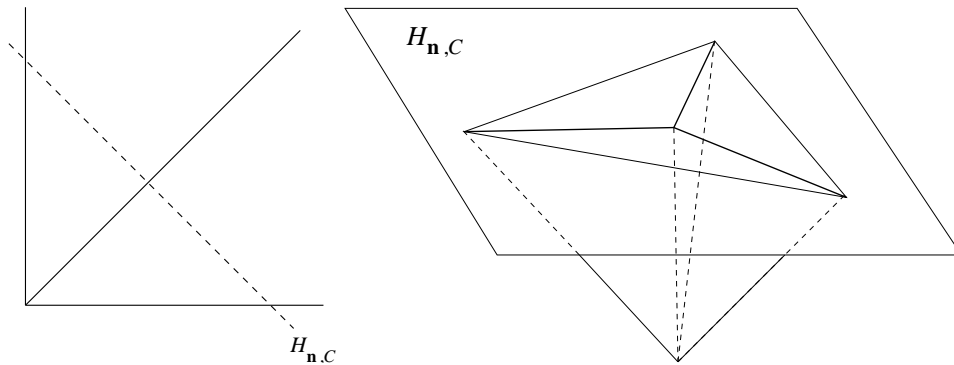


Figure 5.4: Topology of the intersection of the partitioned $(\mathbb{R}_0^+)^n$ with a hypersurface $H = H_{n,C}$, equivalent to the $(n - 2)$ -dimensional surface of a $(n - 1)$ -dimensional (hyper)cube corner. Left: (a) For $n = 3$, two line segments are obtained, equivalent to two sides of a quadrangle. Right: (b) For $n = 4$, one has three triangles, adjacent to a common corner like the corners of three quadrangular facets of a cube.

that of the $(n - 2)$ -dimensional surface of a $(n - 1)$ -dimensional (hyper)cube corner, see Figure 5.4.

We discuss the simplest cases explicitly.

Case $n = 2$. Since B is a scalar, all initial values satisfy the conditions of Proposition 5.14, i.e., no staircasing takes place.

Case $n = 3$. The sets $T_{t<0}(S_1)$ and $T_{t<0}(S_2)$ are separated by a line which due to symmetry considerations and because of the scaling property $\alpha T_{t<0}(S_i) = T_{t<0}(S_i)$ must be the bisector of the quadrant $(\mathbb{R}_0^+)^2$, i.e., $v_1 = v_2$. Initial values with $v_1(0) > v_2(0)$ make v_2 vanish first, others with $v_1(0) < v_2(0)$ make v_1 vanish first.

Case $n = 4$. The three facets S_1, S_2, S_3 bounding the octant $(\mathbb{R}_0^+)^3$ share the boundary half-lines $S_{12} = \{(0, 0, a)^T \mid a \geq 0\}$, $S_{13} = \{(0, a, 0)^T \mid a \geq 0\}$, $S_{23} = \{(a, 0, 0)^T \mid a \geq 0\}$, respectively. Inserting $(0, 0, a)^T$ into (5.63) gives

$$\begin{aligned} v_1(t) &= \frac{1}{2} \left(\frac{1}{2} e^{(2-\sqrt{2})t} - e^{2t} + \frac{1}{2} e^{(2+\sqrt{2})t} \right) a \\ v_2(t) &= \frac{1}{2} \left(\frac{\sqrt{2}}{2} e^{(2-\sqrt{2})t} - \frac{\sqrt{2}}{2} e^{(2+\sqrt{2})t} \right) a \\ v_3(t) &= \frac{1}{2} \left(\frac{1}{2} e^{(2-\sqrt{2})t} + e^{2t} + \frac{1}{2} e^{(2+\sqrt{2})t} \right) a \end{aligned} \quad (5.74)$$

which by the substitution

$$\alpha := \frac{1}{4} e^{2t} (e^{t\sqrt{2}/2} + e^{-t\sqrt{2}/2})^2, \quad \beta := \frac{e^{t\sqrt{2}/2} - e^{-t\sqrt{2}/2}}{e^{t\sqrt{2}/2} + e^{-t\sqrt{2}/2}} \quad (5.75)$$

simplifies to

$$v_1(t) = \alpha \beta^2, \quad v_2(t) = \sqrt{2} \alpha \beta, \quad v_3(t) = \alpha. \quad (5.76)$$

Taking into account that $a > 0$ and $t < 0$, it follows that

$$T_{t < 0}(S_{12}) = \{(\alpha \beta^2, \sqrt{2} \alpha \beta, \alpha)^T \mid \alpha > 0, 0 < \beta < 1\}. \quad (5.77)$$

Analogous considerations lead to

$$\begin{aligned} T_{t < 0}(S_{23}) &= \{(\alpha, \sqrt{2} \alpha \beta, \alpha \beta^2)^T \mid \alpha > 0, 0 < \beta < 1\} \\ T_{t < 0}(S_{13}) &= \{(\alpha, \sqrt{2} \beta, \alpha)^T \mid 0 < \alpha < \beta\} \end{aligned} \quad (5.78)$$

which finally allow us to establish the following set of criteria, for given initial values $v(0) = (v_1(0), v_2(0), v_3(0))^T$:

$$\begin{aligned} v_2(0) < \sqrt{2v_1(0)v_3(0)} &\Rightarrow v_2 \text{ vanishes first,} \\ v_2(0) > \sqrt{2v_1(0)v_3(0)} \text{ and } v_1(0) < v_3(0) &\Rightarrow v_1 \text{ vanishes first,} \\ v_2(0) > \sqrt{2v_1(0)v_3(0)} \text{ and } v_1(0) > v_3(0) &\Rightarrow v_3 \text{ vanishes first.} \end{aligned} \quad (5.79)$$

5.2.2 Numerical Tests

Our experimental validation of the previously obtained results is twofold. On one hand, we validate experimentally the bifurcation results from the preceding paragraph, thus showing that staircasing is predictable by theory. Note, that the test data are chosen so that the experiments featuring staircasing can be understood as perturbed data of the non-staircasing test case, thus showing that data perturbations, e.g. due to low-level noise or preceding numerical errors in the case of FCT schemes, may influence the outcome of a SILD process. Complementing these investigations, we discuss the influence of time stepping schemes by use of a numerical staircase-type solution.

5.2.2.1 Validation of Semidiscrete Theory

To validate the results of our bifurcation analysis, we consider a couple of data segments of length $n = 4$ which can be classified according to Proposition 5.14 and (5.79). As the theoretical results are obtained for the semidiscrete case, we integrate in time using Euler forward time stepping with very small time step sizes, i.e., we generally use $\tau = 10^{-7}$. As in the theoretical derivations, we fix the spatial mesh width to 1.

Case 1: No staircasing. We consider the following set of initial data:

$$(u_0^0, u_1^0, u_2^0, u_3^0, u_4^0, u_5^0) = (2, 1.1707, 1.0707, 0.9293, 0.8293, 0), \quad (5.80)$$

continued by constant states $u_i^0 = 2$, $i < 0$, and $u_i^0 = 0$, $i > 5$, respectively.

Regarding the criteria (5.79), we observe that the data (5.80) correspond to $v_1 = v_3$ and $v_2 = \sqrt{2}v_1$. In Figure 5.5 (top left) we display the initial signal as well as its steady state solution, evaluated at $t = 3$. As predicted, no staircasing effect is visible.

Case 2: Staircasing in the middle of a profile. Consider now the initial data

$$(u_0^0, u_1^0, u_2^0, u_3^0, u_4^0, u_5^0) = (2, 1.16, 1.06, 0.94, 0.84, 0), \quad (5.81)$$

continued as in the preceding test case by constant states left and right. As easily observed, this case corresponds to $v_2 < \sqrt{2}v_1v_3$. Note that the data from (5.81) differ only marginally from signal (5.80). Figure 5.5 (top right) shows the initial signal together with its steady-state solution, evaluated again at $t = 3$. As predicted, staircasing is observable at the middle of the profile, with $u_2^k = u_3^k = 1$ for large k .

Case 3: Staircasing at an end of a profile. Our third set of initial data is

$$(u_0^0, u_1^0, u_2^0, u_3^0, u_4^0, u_5^0) = (2, 1.1707, 1.0807, 0.9393, 0.8293, 0). \quad (5.82)$$

Note again that (5.82) is very close to signal (5.80); however, one easily verifies that in this case $v_2 > \sqrt{2}v_1v_3$ and $v_1 < v_3$.

The initial signal as well as two evolved states are shown in Figure 5.5 (bottom row). At $t = 1.8$, we observe as predicted that staircasing occurs first near the left end of the profile. The steady state solution then is dominated by the first staircasing event.

Remark. The case that v_3 vanishes first, see (5.79), can be realised numerically in an analogous fashion.

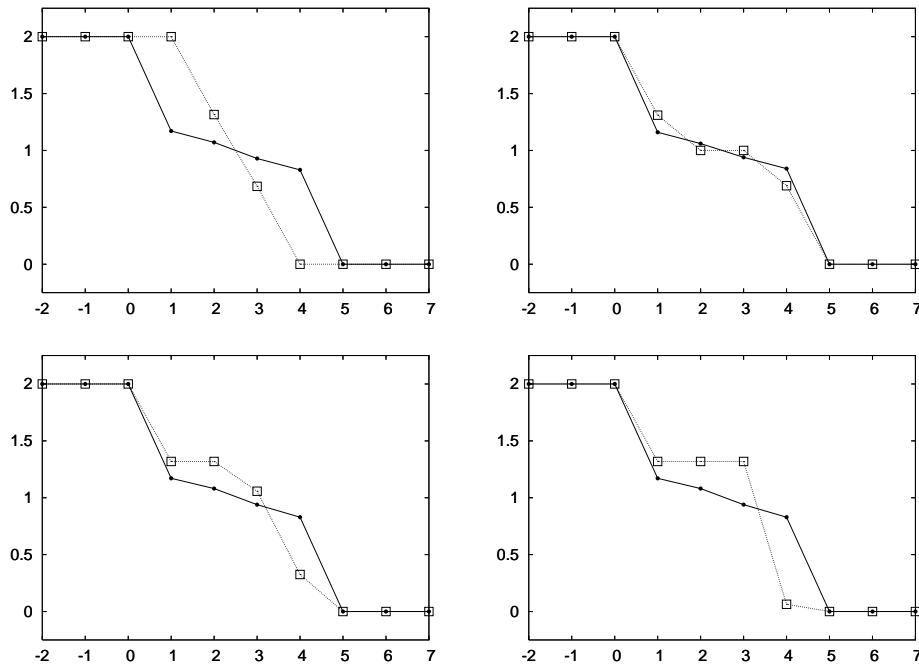


Figure 5.5: Initial states (lines with dots) together with numerical states (squares), as described below. *Top left*: initial signal from (5.80) and steady state without staircasing. *Top right*: initial data from (5.81) and steady state featuring staircasing at the middle of the profile. *Bottom*: initial state from (5.82) together with (*left*) intermediate state with staircasing at an end of the profile, (*right*) steady state dominated by previous staircasing. From [38].

5.2.2.2 Discussion of time integration

In this section, we want to investigate experimentally the influence of time discretisation methods on a stable situation away from a bifurcation situation, thus complementing the above numerical tests.

To this end, we employ a useful academic test case, i.e., The test case we use for this purpose is the *tent function* already suggested in [36], here given as an initial function u_0 :

$$u_0(x) = \begin{cases} \alpha \sin\left(\frac{\pi}{2}\left(\frac{x}{10} + 1\right)\right) + \beta, & -5 \leq x \leq 5 \\ 0 & \text{else.} \end{cases} \quad (5.83)$$

Setting $\alpha = 5$ and $\beta = 0$, we obtain on a grid with spatial mesh width $h = 1$ the function u_0 together with its discrete representation displayed in Figure 5.6 (left).

Taking the discrete data displayed in Figure 5.6 (left) and performing 8 time steps with $\tau = 0.2$ using the *fully discrete* method from (5.36)–(5.38), we obtain the staircasing situation given in Figure 5.6 (right). The discretisation error of the time stepping

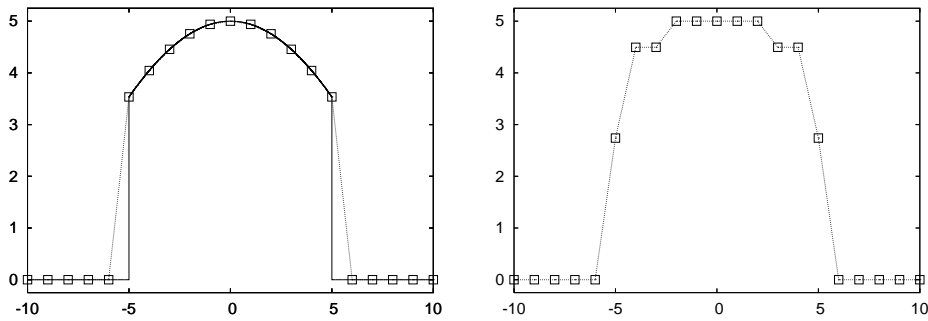


Figure 5.6: *Left*: analytic (line) and discrete (squares) initial states. *Right*: staircasing by propagation of discrete initial state. Adapted from [38].

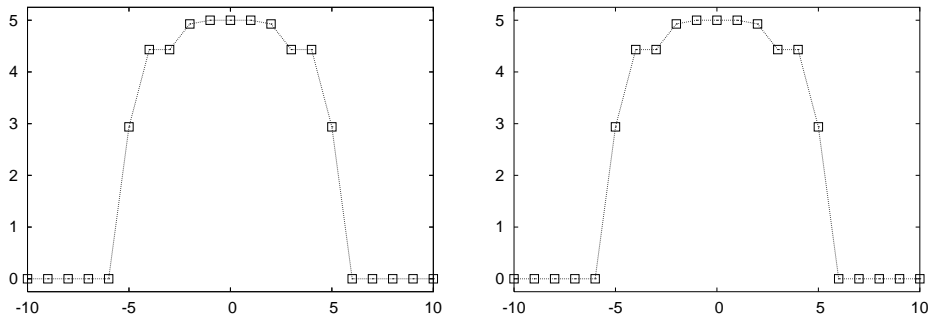


Figure 5.7: Re-computations of staircasing solutions. *Left*: using 160 000 time steps with $\tau = 10^{-7}$. *Right*: using 16 000 000 time steps with $\tau = 10^{-9}$. Adapted from [38].

scheme is $O(\tau)$, such that one obtains in the limit $\tau \rightarrow +0$ the semidiscrete scheme (5.43). For this reason we repeat the computation using the initial data from Figure 5.6 (right) but with very small time step size. The left and right diagrams in Figure 5.7 show the computational results after 160 000 time steps with $\tau = 10^{-5}$, and after 16 000 000 time steps with $\tau = 10^{-7}$, respectively. We observe nearly the same staircase-like structure as in the case of the coarse time discretisation, see Figure 5.6 (right); the differences of the employed time step sizes are observable only by the slightly more rounded structure of the signals in Figure 5.7 compared with Figure 5.6 (right). Here, as staircasing is an unquestioned feature of the spatial discretisation, the error of the time discretisation takes the role of an approximation error resulting in a slightly rougher profile. However, as it is clear after our discussion, staircasing cannot be avoided.

Let us point out a difference between the fully discrete method employing the Euler time stepping method, (5.36)–(5.38), and the semidiscrete methods. In the fully discrete case there exist data constellations circumventing the effect of the minmod stabilisation. To demonstrate this, we modify the case discussed above by choosing $\alpha = 5$ and $\beta = -2.5$, thus translating the tent function from Figure 5.6 (left) moder-

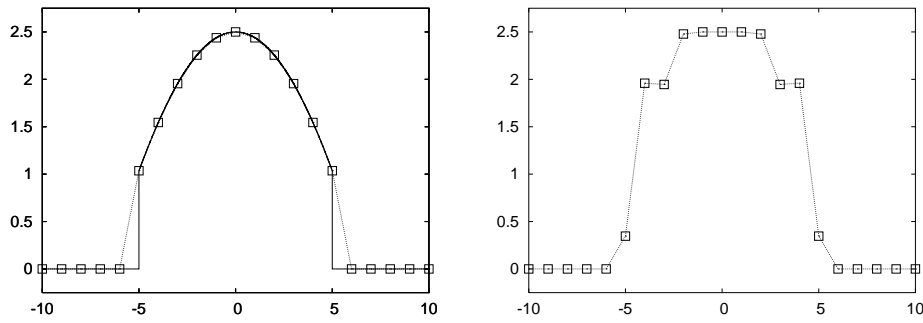


Figure 5.8: *Left*: analytic (line) and discrete (squares) initial states. *Right*: numerical solution (and steady state) after 9 time steps with slight new extrema. Adapted from [38].

ately into negative y -direction, see Figure 5.8 (left). Note the new scaling of the y -axis. Computing 9 time steps with $\tau = 0.001$, we observe that new extrema are produced, see Figure 5.8 (right). The reason for this behaviour is that at the critical points where new extrema arise the method is reduced at the ninth time step to scheme (5.41). The stabilisation then takes no effect, which cannot happen in the semidiscrete case.

It is evident from the last experiment that the fully discrete scheme as stated does not accurately model all important qualitative properties of the semidiscrete process, since the TVP property is violated by the generation of new extrema.

A remedy for this undesired behaviour is to modify the numerical flux (5.38) into

$$g_{i+1/2} = \min\text{mod}\left(\frac{1}{2\tau}(u_{i+2}^n - u_{i+1}^n), c(u_{i+1}^n - u_i^n), \frac{1}{2\tau}(u_i^n - u_{i-1}^n)\right), \quad (5.84)$$

thus restricting the updates of variables within one time step of the fully discrete scheme in such a way that none of two neighbouring pixels is allowed to travel more than half the distance towards its neighbour. This is in fact the same sort of stability limit as used in the 1-D total variation diffusion scheme in 4.1.

A disadvantage from the theoretical point of view could be that neighbouring pixels approach each other only asymptotically, thus postponing the actual merging events from *finite* to *infinite* times. This is the same phenomenon as observed for the semidiscrete shock filter in 5.1, see remark on page 177.

Another way to suppress the TVP violations is based on the observation that it requires two adjacent data moving in opposite directions to generate a new extremum. Transferring therefore the procedure from 5.1.2, page 182, we can design a two-step TVP scheme:

Step 1:

$$\tilde{u}_i^n = u_i^n - \tau(g_{i+1/2} - g_{i-1/2}), \quad (5.85)$$

Step 2:

$$u_i^{n+1} = \begin{cases} \frac{1}{2}(\tilde{u}_{i+1}^n + \tilde{u}_i^n), & (\tilde{u}_{i+1}^n - \tilde{u}_i^n)(u_{i+1}^n - u_i^n) < 0, \\ \frac{1}{2}(\tilde{u}_{i-1}^n + \tilde{u}_i^n), & (\tilde{u}_{i-1}^n - \tilde{u}_i^n)(u_{i-1}^n - u_i^n) < 0, \\ \tilde{u}_i^n & \text{else.} \end{cases} \quad (5.86)$$

These steps substitute (5.36), while (5.38) is retained without change. Note, that the modification by (5.85)–(5.86) is conservative as data at (automatically adjacent) new extrema are replaced by their average.

Both schemes, (5.36) with (5.84) as well as (5.85)–(5.86) with (5.38), are time-discrete approximations for the semidiscrete process (5.43) since all modifications vanish as τ goes to zero.

5.3 Forward-and-Backward Diffusion

In this section we are concerned with a fully discrete analysis of the diffusion process

$$u_t = \operatorname{div}(g(|\nabla u|^2) \nabla u) \quad (5.87)$$

where the diffusivity g is positive at zero, but can attain negative values for some positive arguments. In detail we require that

$$g(0) = c_1 \quad (5.88)$$

$$g(s^2) \in [c_2, c_1] \quad \text{for all } s > 0 \quad (5.89)$$

with constants c_1, c_2 that satisfy

$$|c_1| > |c_2|, \quad c_1 > 0 > c_2. \quad (5.90)$$

We will also demand that there exists a small interval $[0, z^2)$ with $z > 0$ in which all values of g are positive. For continuous g this is automatically satisfied.

This type of diffusion process is called *forward-and-backward diffusion* [95].

Specialising to the 1D case, we obtain

$$u_t = \partial_x (g(u_x^2) u_x). \quad (5.91)$$

Spatial Discretisation. A spatial discretisation of (5.91) with spatial mesh width 1 yields a semi-discrete model which reads

$$\dot{u}_i = \frac{g_{i-1} + g_i}{2} (u_{i-1} - u_i) + \frac{g_{i+1} + g_i}{2} (u_{i+1} - u_i) \quad (5.92)$$

where we have discretised the diffusivity g in the grid points by

$$g_i = g(\max((u_i - u_{i-1})(u_{i+1} - u_i), 0)). \quad (5.93)$$

Comparing to the previous section, we see that this discretisation again involves a stabilisation. The use of a mixed product of one-sided derivatives in the argument of g ensures that we detect discrete extrema of u correctly: In these points, $(u_i - u_{i-1})(u_{i+1} - u_i)$ is zero or negative (despite the fact that u_x^2 cannot become negative) and is mapped to zero by the maximum operation. This is similar to the minmod discretisation of the previous section. However, the averaging of neighbouring diffusivity values as in $\frac{g_i + g_{i+1}}{2}$ implies that we do this time not force the fluxes adjacent to an extremum to be zero. It will turn out that with our requirements to g we can still ensure maximum–minimum stability.

Time-Discrete Model. Discretising also the time variable, we arrive at a fully discrete model. It takes the form

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \frac{g_{i-1}^k + g_i^k}{2} (u_{i-1}^k - u_i^k) + \frac{g_{i+1}^k + g_i^k}{2} (u_{i+1}^k - u_i^k) \quad (5.94)$$

with the time step size τ and $g_i^k = g(\max((u_i^k - u_{i-1}^k)(u_{i+1}^k - u_i^k), 0))$.

5.3.1 Stability and Well-Posedness Analysis

For our analysis, two additional assumptions are essential. While the first one refers to the range of grey-values, the second one requires that the diffusivity g still takes sufficiently large positive values for small positive arguments. We can state the following result.

Theorem 5.16 *Let an initial 1-D image $f = (f_i)$ be given and let the sequence of images $u^k = (u_i^k)$ evolve according to (5.94) with the initial condition $u^0 = f$. Let the grey-values f_i be restricted to a finite interval of length R . Assume further that for the diffusivity g , a positive ω exists such that $g(s^2) > -c_2$ holds for all s , $0 < s < \omega R$.*

If the time step satisfies

$$\tau < \frac{\omega^2}{c_1 + |c_2| + 2c_1\omega^2}, \quad (5.95)$$

the following are true for the time evolution (u^k) .

- (i) **Maximum–minimum principle.** *If the initial signal is bounded by $a \leq f_i \leq b$ for all $i = \dots, -1, 0, 1, 2, \dots$, then the inequalities $a \leq u_i^k \leq b$ hold for all i and all $k = 0, 1, 2, \dots$*
- (ii) **Total variation reduction.** *For each time step $k = 0, 1, 2, \dots$, the total variation of the image u^{k+1} is less or equal to the total variation of u^k .*

Proof. The global statements of the theorem follow from local properties which will be proved in the sequel.

1. We prove that a local maximal pixel does not increase. Assume that u_i^k is a local maximum of the 1-D image in time step k , i.e., we have $u_i^k \geq u_{i+1}^k$ and $u_i^k \geq u_{i-1}^k$. Since in this case $g_{i-1}^k + g_i^k$ and $g_i^k + g_{i+1}^k$ are certainly nonnegative, u_i^{k+1} is a convex combination of u_i^k , u_{i+1}^k and u_{i+2}^k if only

$$1 - \frac{\tau}{2}(g_{i-1}^k + 2g_i^k + g_{i+1}^k) \geq 0 \quad (5.96)$$

holds. Because of $g_{i-1}^k + 2g_i^k + g_{i+1}^k \leq 4c_1$ this is surely the case if

$$\tau \leq \frac{1}{2c_1}. \quad (5.97)$$

2. We show that the neighbour pixel of a local maximum can not increase in excess of this maximum. Assume that u_i^k is a maximum and u_{i+1}^k is not a local minimum. Then the inequality $u_{i+1}^{k+1} \leq u_i^k$ holds if

$$\tau \leq \frac{\omega^2}{2c_1\omega^2 + |c_2|}. \quad (5.98)$$

To see this, we use the equation

$$u_{i+1}^{k+1} = u_{i+1}^k + \tau \cdot \left(\frac{g_i^k + g_{i+1}^k}{2}(u_i^k - u_{i+1}^k) + \frac{g_{i+1}^k + g_{i+2}^k}{2}(u_{i+2}^k - u_{i+1}^k) \right) \quad (5.99)$$

and distinguish two cases.

Case 1: $(u_{i+1}^k - u_i^k)(u_{i+2}^k - u_{i+1}^k) \leq \omega^2 R^2$.

Then $g_{i+1}^k + g_{i+2}^k$ is certainly nonnegative. The right-hand side of (5.99) is therefore a convex combination of u_i^k , u_{i+1}^k and u_{i+2}^k if

$$1 - \frac{\tau}{2}(g_i^k + 2g_{i+1}^k + g_{i+2}^k) \geq 0. \quad (5.100)$$

Analogous to our above reasoning, this is true if (5.97) is satisfied.

Case 2: $(u_{i+1}^k - u_i^k)(u_{i+2}^k - u_{i+1}^k) > \omega^2 R^2$.

Here we conclude from $u_{i+1}^k - u_{i+2}^k \leq R$ that

$$u_i^k - u_{i+1}^k > \omega^2 R. \quad (5.101)$$

Using $\frac{1}{2}(g_i^k + g_{i+1}^k) < c_1$ and $\frac{1}{2}(g_{i+1}^k + g_{i+2}^k) > c_2$ we obtain from (5.99) the estimate

$$u_{i+1}^{k+1} \leq u_{i+1}^k + \tau c_1 (u_i^k - u_{i+1}^k) + \tau |c_2| R \quad (5.102)$$

which ensures $u_{i+1}^{k+1} \leq u_i^k$, provided that

$$\tau \leq \frac{\omega^2}{c_1 \omega^2 + |c_2|} \quad (5.103)$$

holds.

Condition (5.98) ensures the bounds of both cases, i.e., (5.97) and (5.103).

3. We prove that no new extrema are generated around existing extrema. Assume that u_i^k is a local maximum, and none of its neighbours is a local minimum. Assume first that

$$(u_{i+1}^k - u_i^k)(u_{i+2}^k - u_{i+1}^k) > \omega^2 R^2 \quad (5.104)$$

and thus again (5.102) holds.

Similar considerations for u_i^{k+1} yield

$$u_i^{k+1} \geq u_i^k + \tau c_1 (u_{i+1}^k - u_i^k) - \tau c_1 R \quad (5.105)$$

and therefore

$$u_i^{k+1} - u_{i+1}^{k+1} \geq (1 - 2\tau c_1)(u_i^k - u_{i+1}^k) - \tau(c_1 + |c_2|)R. \quad (5.106)$$

Since we assume that

$$\tau < \frac{\omega^2}{c_1 + |c_2| + 2c_1 \omega^2} \leq \frac{1}{(c_1 + |c_2|)R / (u_i^k - u_{i+1}^k) + 2c_1}, \quad (5.107)$$

the expression on the right-hand side of (5.106) is nonnegative.

Therefore u_{i+1}^{k+1} can become a maximum in (u^{k+1}) only if

$$(u_{i+1}^k - u_i^k)(u_{i+2}^k - u_{i+1}^k) \leq \omega^2 R^2. \quad (5.108)$$

Analogous reasoning applies to the left neighbour u_{i-1}^{k+1} . This means that the maximum property of pixel i can be shifted to one of its neighbours. Our assertion that no new extrema are generated remains true except if both neighbours u_{i-1}^{k+1} and u_{i+1}^{k+1} would simultaneously into maxima.

Let us therefore discuss this case. This would require the two inequalities

$$\begin{aligned} (u_{i+1}^k - u_i^k)(u_{i+2}^k - u_{i+1}^k) &\leq \omega^2 R^2 \\ \text{and } (u_{i-1}^k - u_{i-2}^k)(u_i^k - u_{i-1}^k) &\leq \omega^2 R^2 \end{aligned} \quad (5.109)$$

to hold at the same time. In this situation, however, $g_{i+1}^k + g_{i+2}^k$ and $g_{i-1} + g_{i-2}$ are nonnegative, implying

$$\begin{aligned} u_{i+1}^{k+1} &\leq u_{i+1}^k + \tau c_1 (u_i^k - u_{i+1}^k) \\ \text{and } u_{i-1}^{k+1} &\leq u_{i-1}^k + \tau c_1 (u_i^k - u_{i-1}^k), \end{aligned} \quad (5.110)$$

while for the central pixel

$$u_i^{k+1} \geq u_i^k + \tau c_1 (u_{i-1}^k - 2u_i^k + u_{i+1}^k) \quad (5.111)$$

holds, hence

$$-u_{i-1}^{k+1} + 2u_i^{k+1} - u_{i+1}^{k+1} \geq (1 - 2\tau c_1)(-u_{i-1}^k + 2u_i^k - u_{i+1}^k). \quad (5.112)$$

For $\tau \leq \frac{1}{2c_1}$, the right-hand side is clearly nonnegative which ensures that u_{i-1}^{k+1} and u_{i+1}^{k+1} can not both become maxima.

4. We prove that monotonicity is preserved in image segments without extrema. Assume that $u_i^k > u_{i+1}^k > u_{i+2}^k > u_{i+3}^k$. We show that then also $u_{i+1}^{k+1} \geq u_{i+2}^{k+1}$ holds.

In the proof we distinguish three cases.

Case 1: $g_i^k + g_{i+1}^k \geq 0$ and $g_{i+2}^k + g_{i+3}^k \geq 0$.

Then

$$u_{i+1}^{k+1} - u_{i+2}^{k+1} \geq (1 - 2\tau c_1)(u_{i+1}^k - u_{i+2}^k); \quad (5.113)$$

the right-hand side is again surely nonnegative if (5.97) holds.

Case 2: $g_i^k + g_{i+1}^k \geq 0$ and $g_{i+2}^k + g_{i+3}^k < 0$.

(The case $g_i^k + g_{i+1}^k < 0$ and $g_{i+2}^k + g_{i+3}^k \geq 0$ is treated in a symmetric way.)

Here we conclude from $u_{i+2}^k - u_{i+3}^k \leq R$ and $(u_{i+1}^k - u_{i+2}^k)(u_{i+2}^k - u_{i+3}^k) > \omega^2 R^2$ that

$$u_{i+1}^k - u_{i+2}^k > \omega^2 R. \quad (5.114)$$

Consequently,

$$\begin{aligned} u_{i+1}^{k+1} - u_{i+2}^{k+1} &\geq u_{i+1}^k - u_{i+2}^k - 2\tau c_1 (u_{i+1}^k - u_{i+2}^k) - \tau |c_2| (u_{i+2}^k - u_{i+3}^k) \\ &> u_{i+1}^k - u_{i+2}^k - 2\tau c_1 (u_{i+1}^k - u_{i+2}^k) - \tau |c_2| R; \end{aligned} \quad (5.115)$$

the right-hand side is certainly nonnegative if

$$\tau \leq \frac{\omega^2}{2c_1 \omega^2 + |c_2|}. \quad (5.116)$$

Case 3: $g_i^k + g_{i+1}^k < 0$ and $g_{i+2}^k + g_{i+3}^k < 0$.

Since in this case we have

$$(u_i^k - u_{i+1}^k) + (u_{i+2}^k - u_{i+3}^k) \leq R, \quad (5.117)$$

it follows that

$$(u_{i+1}^k - u_{i+2}^k) \min(u_i^k - u_{i+1}^k, u_{i+2}^k - u_{i+3}^k) > \omega^2 R^2 \quad (5.118)$$

and thus

$$u_{i+1}^k - u_{i+2}^k > 2\omega^2 R. \quad (5.119)$$

Similar reasoning as in Case 2 gives that $u_{i+1}^{k+1} - u_{i+2}^{k+1}$ is ensured if

$$\tau \leq \frac{\omega^2}{2c_1\omega^2 + |c_2|/2}. \quad (5.120)$$

Comparing the bounds derived for the different statements yields (5.95) as the most restrictive one, which we impose therefore. Since then extrema can not be generated but only shifted to neighbouring pixels, and monotonic segments preserve their monotonicity, both maximum–minimum principle and reduction of total variation follow immediately. \square

Remark. Theorem 5.16 demonstrates that in the semidiscrete setting even a process involving negative diffusivities can have well-posedness properties, which can be analysed by our approach. In applying this theoretical result for practical computation, however, one is confronted with the difficulty that the bound for τ guaranteed by the Theorem is rather small. We do not address this aspect further here.

Part IV

Variational Image Deconvolution

Chapter 6

Variational Models and Integro-Differential Equations for Image Deconvolution

In this and the following chapter, we are concerned with the application of variational approaches and related (integro-)differential equations for the sharpening of images. The present chapter is devoted to our basic model. The relevant concepts and approaches have been laid out in two conference papers at Scale Space 2005 [222] and DAGM 2005 [223]. The presentation here follows and extends these two sources. The author is particularly grateful to David Theis who contributed to this research within his diploma thesis [193] under the author's co-supervision, and did many experiments to validate and compare concepts, and who is also a coauthor of the before-mentioned conference papers.

6.1 Introduction to Basic Deblurring Concepts

Blurring describes the fact that information belonging to a single object point depicted in an image is spread over a certain region in the image, instead of being sharply localised. In all kinds of image acquisition methods, this is a ubiquitous phenomenon. Common sources of blurring include

- motion of objects during the recording of an image,
- motion of the camera during the recording of an image,
- defocussing of the imaging equipment,

- aberrations, diffraction and other deviations of the real camera optics from its idealised model,
- atmospheric perturbations.

Consequently, there is a strong demand for methods which can remove blur, i.e., *deblur* images, in order to fully exploit the information contained in digital images.

Deblurring, or image restoration, as it is also often called, is a challenging, severely ill-posed problem. It is therefore reasonable to introduce into the deblurring process any a-priori knowledge available in a specific application context in order to improve the quality of results. Thus it is natural that over the years a tremendous variety of approaches has been proposed in the literature, which differ greatly in their assumptions on the image as well as on the blurring process, and in the way how these assumptions are cast into a mathematical model.

Before we give a coarse categorisation of deblurring approaches, let us establish some fairly general mathematical framework.

6.1.1 Basic Blur Model

We have to describe the relation between the *observed image* $f : \Omega \rightarrow R$ given on a (discrete or continuous) image domain Ω and the “ideal” image $g : \Omega' \rightarrow R$ on the image domain Ω' . The latter is often identified with the physical object being imaged. We assume that both f and g share the same range R which is a real vector space, but we permit for the moment that the image domains Ω and Ω' may be different.¹

A widespread and physically well motivated assumption in deblurring methods is that the blurring process is a linear process that can be modelled as

$$u(x) = \int_{\Omega'} g(y)H(x,y)dy \quad (6.1)$$

with a function $H : \Omega \times \Omega' \rightarrow \mathbb{R}$. The latter is called *point-spread function (PSF)* since it describes the contributions of each point y of the ideal image domain Ω' to each point x of the observed image domain Ω . In cases when the image domain Ω' considered for g is discrete, the integral over Ω' is of course a sum.

¹Although we will not make substantial use of this generality in this work, it is useful in several respects for further developments. We sketch two directions. Firstly, a model with $\Omega \neq \Omega'$ can better account for the fact that the blurring transports information from outside to inside the observed image region. We will discuss ways to handle these boundary effects later in this chapter. Secondly, one could even choose e.g. Ω' as a continuous domain but Ω discrete in order to model the entire image acquisition process from the sharp continuous physical image to the blurred discrete observed image. Models of this type, however, are not considered further here.

However, realistic models have to account for the fact that measurements are noisy. This means that instead of the right-hand side of (6.1) one observes the result of some stochastic point operation $N_{(\cdot)}(\cdot) : \Omega \times R \rightarrow R$ acting on it, i.e.,

$$f(x) = N_x \left(\int_{\Omega'} g(y)H(x,y) dy \right). \quad (6.2)$$

Usually, the noise operator N is assumed to be additive, such that the model simplifies to

$$f(x) = \int_{\Omega'} g(y)H(x,y) dy + n(x) \quad (6.3)$$

with a random noise function $n : \Omega \rightarrow R$.

6.1.2 Classification of Deblurring Approaches

We will now sketch three main distinctions that can be used to classify deblurring approaches.

Type of blurring. An important case of image blurring is that the values of the PSF depend only on the relative position of the two points involved. In this case one speaks of a *spatially invariant PSF*, otherwise of a *spatially variant PSF*.

The spatially invariant blurring model tacitly assumes that the image domains Ω and Ω' can be identified, and possess an affine structure, such that $x - y$ makes sense for $x \in \Omega, y \in \Omega'$. It requires then that $H(x,y) = h(x-y)$.² One has then

$$f(x) = \int_{\Omega} g(y)h(x-y) dy + n(x) \quad (6.4)$$

which can easily be rewritten as

$$f = g * h + n \quad (6.5)$$

where $*$ denotes convolution.

From this case, it is clearly motivated why the term *deconvolution* is often used in place of *deblurring*. It has, however, become common to denote by deconvolution also the solution of more general integral equations, which justifies the use of the term also for deblurring with spatially variant PSF.

Figure 6.1 shows examples of spatially invariant point-spread functions.

²This assumption can be weakened.

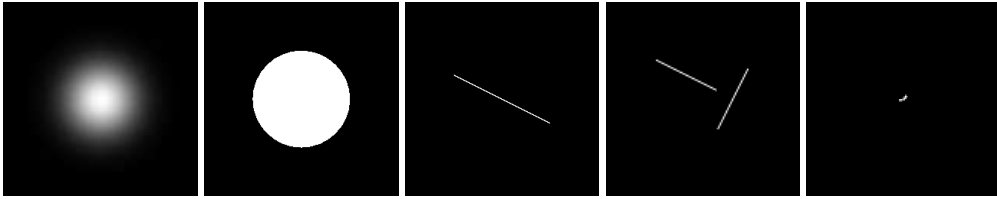


Figure 6.1: Some spatially invariant point-spread functions (convolution kernels). *Left to right:* (a) Gaussian. It can be used as simple model for different types of blur, e.g. as a coarse approximation for atmospheric blur. (b) Disk-shaped PSF. It appears typically as defocussing kernel for rotationally symmetric apertures. (c) Line-shaped PSF as it results from linear motion blur. (d) Discontinuous PSF combined from two line-shaped components. (e) Banana-shaped PSF, extracted from a photograph where it appeared as impulse response of an approximate point light source (compare Figure 6.8). – Some of these PSFs will be used in experiments later in this chapter.

Comparing to general spatially variant PSFs, spatially invariant PSFs contain tremendously less data. Given two-dimensional images, e.g., a full space-variant PSF has a four-dimensional domain of definition. Storing it completely would require huge memory resources even for moderate image sizes and moderate reach of blur and is therefore unpractical. Besides, it is unrealistic that such a fully general space-variant PSF could be measured. For these reasons it is fairly attractive, and common practice, to work with space-invariant PSF models even when they are known to be inexact as long as the inaccuracy is moderate.

Knowledge about the blurring process. The second distinction to be made refers to the information that is available on the PSF. Some methods are designed to deblur images where the point-spread function H is known. Such knowledge can be derived from measurements, for example from calibration of camera optics, or from theoretical considerations of the physical process that leads to blurring, or from a combination of both. We will refer to these approaches as models with *known blur*.

As opposed to that, *blind deblurring methods* try to recover even the parameters of blurring from the blurred image. Obviously, a blind deblurring problem is much more underdetermined than one with known blur. Inevitably, even blind deblurring methods have to make some assumptions on the type of blur, which in fact leads to a scale of degrees of blindness. On one extreme, fully blind deblurring with spatially variant PSF is seems out of reach due to its extreme underdetermination. On the other extreme, in some cases, where the search space is restricted to a blur model with very few parameters, one speaks of *semi-blind* methods. Clearly, the existence of blind deblurring methods does not render non-blind approaches superfluous.

Discretisation level. The third distinction refers to the image domains Ω, Ω' used in designing the algorithm: They can be chosen either discrete or continuous.

Since measured digital images are necessarily on a discrete domain, discrete models are closer to the image representation that is finally used in the numerical computations, while continuous models still need to be discretised for implementation. Moreover, discrete models can capitalise on the realm of matrix algebra theory; alternatively, discrete variational approaches can be formulated.

On the other hand, continuous modelling is closer to the physical process of blurring that takes place in a continuous environment. It allows to use the theory of (integro-) differential equations and closely related calculus of variations. This framework ensures by construction that important physical symmetries are properly reflected in the model.

Hence, the choice between continuous and discrete modelling appears primarily as a methodological decision. Both approaches have led to promising results, and both approaches display specific strengths. It is therefore worth trying to combine concepts developed in both frameworks.

6.1.3 Related Work

Deblurring problems have attracted the attention of signal processing and computer vision researchers for almost six decades. Early contributions were based on linear system theory. Later on, the accent of linear methods shifted towards matrix-algebraic methods which profit from the great developments in iterative solvers such as conjugated gradients, particularly with preconditioning. Variational approaches to non-blind and blind deconvolution date back mostly to the last 15 years.

Linear filtering. A first class of approaches to deconvolution within a continuous framework (with spatially invariant kernel) is based on ideas from linear system theory. Assume that a blurred image is given which is the convolution of a sharp image with a known kernel superposed with additive noise,

$$f(x) = (g * b)(x) + n(x), \quad x \in \mathbb{R}^2. \quad (6.6)$$

Assume first that the noise n can be neglected. Fourier transform then carries over the convolution to multiplication in the frequency domain,

$$\hat{f} = \hat{g} \cdot \hat{b}. \quad (6.7)$$

Provided b is known, division of \hat{f} by \hat{b} should allow to restore \hat{g} and thereby g . However, this *inverse filtering* procedure faces the problem that in general \hat{b} will possess zeroes. These represent frequencies which are deleted by blurring with b and

must therefore not be present in a noise-free blurred image. But still \hat{h} is close to zero in the vicinity of its zeroes, and, even if it has no zeroes, for high frequencies. But in frequency ranges where $|\hat{h}|$ is small, even minimal amounts of noise are tremendously amplified, rendering the procedure extremely unstable.

The simplest approach to handle this difficulty is *pseudo-inverse filtering* which eliminates frequencies for which \hat{h} is smaller than some threshold K . A refined way to regularise inverse filtering is *Wiener filtering* [231] which replaces \hat{h}^{-1} by a smooth approximation $\hat{h}^{-1} |\hat{h}|^2 / (|\hat{h}|^2 + K^2)$ such that one obtains an approximation u for the sharp image g via

$$\hat{u} = \frac{1}{\hat{h}} \cdot \frac{|\hat{h}|^2}{|\hat{h}|^2 + K^2} \cdot \hat{f}. \quad (6.8)$$

This filter displays better stability than pseudoinverse filtering. It has properties of a band-pass and is therefore even well-suited to deal with moderate noise. The optimal value for the threshold parameter K can be linked to the noise level in the image. However, the cut-off of high frequencies still leads to characteristic perturbations in the image which can principally not be avoided by linear filters, see [23, p. 119].

Analytic approaches to discrete deblurring. A number of approaches to discrete deconvolution with Gaussian kernels is based on different analytic approximations of the inverse of Gaussian convolution. Approximate convolution inverses – i.e., convolution kernels providing for an approximative deblurring – of Gaussian PSFs were constructed using Hermite polynomials in [109]. Analytic inversion of the Toeplitz matrix describing Gaussian convolution has been used in [119]. Higher order derivatives computed in the Fourier domain are used in [192]. An approach via q -series was proposed in [135]. An approach integrating analytic inversion with variational ideas can be found in [49].

Iterative and algebraic discrete deblurring methods. Early iterative methods for discrete deblurring are described in the survey paper [26].

More recently, a number of important contributions to discrete deblurring has been made by Nagy and colleagues. Here, spatially invariant PSFs have been considered in [114, 115, 148]. Kronecker product approximations and the theory of Toeplitz (and closely related) matrices are the main approaches. Extensions to spatially variant blurs were considered in [113, 149, 150]. It is assumed that the PSF does not vary largely within small regions such that locally spatially invariant deblurring can be used. Approximate convolution kernels at specific locations are obtained by interpolation.

In [105, 151] it has been demonstrated how to include nonnegativity constraints into a discrete deblurring approach. Boundary conditions were discussed in [148].

The aforementioned methods are applied to astronomic imaging in [104]. An application to the cosmic microwave background has been demonstrated for symmetric [201] and asymmetric [202] PSFs.

Variational deconvolution. Variational methods rely on stating the problem to be solved as minimisation of a functional dependent on an unknown function. Such approaches also have a long tradition in image deblurring. Similar to other variational functionals used in image analysis applications, variational functionals for deblurring are composed of a data (or fidelity) term and a regularisation term. A typical functional has the form

$$E(u) = \int_{\Omega} ((h * u - f)^2 + \Psi(|\nabla u|^2)) \, dx . \quad (6.9)$$

Here, the data term $\int (h * u - f)^2 \, dx$ is based on a mean-square error of the reconstruction of the blurred image f from the deblurred image candidate u and convolution kernel h and enforces the exact match between the blurred image and its deblurred counterpart. The regularisation term $\int \Psi(|\nabla u|^2) \, dx$ encodes a smoothness prior for the deblurred image. This basic structure of a variational formulation of deblurring is found e.g. in [14, 59, 138]. Regularisation ideas in connection with iterative solution are already found in [26, 129]. Research on existence, uniqueness, and stability of solutions for this type of problems can be found in the work by Bertero, Poggio, and Torre [24].

Gradient descents for functionals of this type are represented by integro-differential equations. These resemble diffusion–reaction PDEs but involve convolutions in the reaction term which is therefore not strictly local. Meanwhile, via the diffusion term a link is established to the well-developed theory of image processing via diffusion, see Chapter 2. Many different diffusivities can be linked to different regularisers in variational functionals. The simplest possible regularisation, a quadratic Tikhonov regulariser, generates a linear diffusion component which leads like in other image processing applications to a general degradation of small-scale image details, acting thereby against the desired deblurring effect. Therefore the non-quadratic regularisers which have been established as a standard in image processing for a long time, are also advocated in the variational deblurring literature.

The most widespread nonquadratic regularisation employed in variational deblurring is the total variation (TV) regularisation, see [154, 157], which is linked to total variation diffusion, compare 2.2.4.1. It has been used by Marquina and Osher [138] for non-blind deconvolution. Chan and Wong [59] attacked the blind deconvolution problem with the same regulariser. Another early approach to blind variational deconvolution with quadratic regularisation is found in You and Kaveh [233, 234], where also alternating gradient descent for image and convolution kernel is proposed to carry out the minimisation.

Recent work by Bar, Sochen and Kiryati [14] uses instead a Mumford-Shah segmentation term [147] to address non-blind deconvolution with fairly general spatially invariant PSFs as well as a semi-blind deconvolution problem with a parametric Gaussian PSF model.

Robust data terms are motivated by robust statistics [103, 108] and have been introduced in computer vision particularly in motion detection [28, 29, 40, 107, 141]. Important theoretical results were contributed by Nikolova [152]. In the context of discrete deconvolution models, robust data terms have already been proposed in 1995 by Zervakis et al. [235], but it seems that this proposal attracted little attention at that time. They did not come into broader use until recently, when they were introduced in the variational deblurring framework by Bar et al. [15], where their excellent capabilities in the deconvolution of images degraded by severe, e.g. impulsive, noise were impressively demonstrated. Since then, further aspects of this concept have been investigated also in our own work [223] including the formulation of the spatially variant setup, and in [13] where the focus lies on colour images.

Continuation strategy. In 6.4.3.2, we describe a continuation strategy for the steering of the regularisation weight. This technique has been considered for non-convex variational problems in visual reconstruction in general by Blake and Zisserman [30], and more specifically in the context of total variation denoising by Chan, Chan and Zhou [58].

Blur estimation. A bridge between nonblind and blind deblurring methods is established by several works concerned with estimating point-spread functions.

Again, these methods have to rely on assumptions made about the type of PSF according to which they can be classified. Older approaches [26, 177, 178, 179] are based on parametric models for spatially invariant PSFs; image information is then exploited to estimate parameters.

Recently, Fergus et al. [89] proposed a statistics-based method to estimate irregularly shaped spatially invariant PSFs. Other recent extensions [12, 16, 84, 132, 190] aim at estimating spatially variant PSFs which can be represented by parametric models with spatially variant parameters, like spatially variant defocus or motion blur.

6.1.4 Our Contributions

In this chapter we present a fairly general framework for non-blind deblurring that is based on a variational approach. Via the variational gradient descent procedure it leads to integro-differential equations for deblurring. We list some main features of the approach.

Generality of point-spread functions. The model is designed to impose only weak conditions on the PSFs. Throughout this chapter, we adhere to a spatially variant blur model. No symmetry assumptions (like axial or rotational symmetry) are made such that PSFs can take fairly irregular shapes.

Edge preservation and enhancement. By the choice of regularisers (or corresponding diffusion expressions in the resulting integro-differential equations) the model is enabled to preserve and even enhance edges in images. To this end, we include in the framework besides the common total variation regulariser also non-convex regularisers of Perona–Malik type. On the level of the integro-differential equations we also investigate anisotropic diffusion expressions for which no corresponding energy functional is known so far. The performance of different regularisers is compared quantitatively and visually.

Robustness. We use the setup for nonquadratic robust data terms that has also been used in [15], which is related to the earlier discrete formulation of [235]. With regard to the capabilities of the model, this may be the most valuable single improvement.

The capability of robust data terms of handling extreme noise in image deconvolution is known from Bar et al.'s work [15]. We investigate further practical aspects of this concept. Especially it turns out that a model with robust data terms can excellently cope with a variety of model violations, i.e., discrepancies between idealised model assumptions and the features present in real data.

Although blind deconvolution is not included in the model as presented here, the aforementioned robustness property opens a way to cope with uncertainty on the PSF: Since a misestimation of the blurring PSF is in fact a model violation, robust data terms allow to tolerate it as long as the error is moderate. Thereby, deblurring can be performed even with only approximative knowledge of the PSF, which can be achieved in multiple ways.

Steering of the regularisation/diffusion weight. Based on considerations on the twofold role of the regularisation, namely that it provides stabilisation against noise influence as well as suppression of oscillatory artifacts in the reconstructed image, we propose a continuation strategy that successively reduces the weight parameter.

Boundary conditions. Information flow across the image boundary appears naturally in image blurring and makes the boundary treatment in deblurring more difficult than in other image processing tasks. We discuss shortly some ways to handle this problem. We sketch a setup for synthetic examples which allows quantitative measurements of reconstruction accuracy without the influence of boundary effects.

For practical applications, we propose a pragmatic solution that once more makes use of the favourable robustness properties achieved by nonquadratic data terms.

6.2 Energy Functionals and Diffusion-Reaction Equations

Variational deconvolution methods aim at reconstructing the sharp image by minimising an energy functional that encodes the convolution relation between the given and sought images together with regularity assumptions on the sought image.

6.2.1 Energy Functionals and Diffusion-Reaction Equations

From now on, we simplify the blur model (6.3) by the assumption $\Omega = \Omega'$. Relying on this blur model, deconvolution of an image can be achieved by minimising the energy functional

$$E[u] := E_1[u] + \alpha E_2[u] \quad (6.10)$$

consisting a *data* or *fidelity term* E_1 and a *regularisation* or *smoothness term* E_2 as follows, whose respective influence is balanced by the *regularisation* or *smoothness weight* $\alpha > 0$.

Data term. The data term has the structure

$$E_1[u] := \frac{1}{2} \int_{\Omega} \Phi((R_{f,H}[u](x))^2) dx \quad (6.11)$$

and depends on the *residual*

$$R_{f,H}[u](x) := f(x) - \int_{\Omega} u(y)H(x,y) dy \quad (6.12)$$

that measures the error of reconstruction of the observed image from the estimated sharp image u under our blur model. The function $\Phi : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is an increasing penaliser with $\Phi(0) = 0$. Relevant choices for Φ will be discussed in Section 6.3.1. This setting for the data term comprises and extends its counterparts used in [234, 14] and other variational deconvolution literature. Its purpose is to enforce closeness of the reconstructed image to the unobservable sharp image g from (6.3).

Regulariser. The regularisation term has the structure

$$E_2[u] := \frac{1}{2} \int_{\Omega} \Psi(|\nabla u|^2) dx \quad (6.13)$$

where $\Psi : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is again an increasing penaliser with $\Psi(0) = 0$, for which we will present possible choices in Section 6.3.2. This term is used to enforce smoothness assumptions on the deconvolved image and is therefore the primary place to introduce prior knowledge about image structure into the framework.

The role of the regulariser in variational deconvolution will be discussed further in Section 6.4.3.1.

6.2.2 Variational Gradient Descent

In the following, we aim at finding local minima of E by gradient descent. To this end, we will derive now integro-differential equations that encode this gradient descent. The variational derivation is similar to Chapter 2.

Variational gradient of the data term E_1 . We substitute $w = u + \varepsilon v$ for u into E_1 , where v is assumed to be a smooth function on Ω which vanishes on the boundary. It follows that

$$\begin{aligned}
\left. \frac{d}{d\varepsilon} E_1[w] \right|_{\varepsilon=0} &= \left. \frac{d}{d\varepsilon} \left(\frac{1}{2} \int_{\Omega} \Phi((R_{f,H}[w](x))^2) dx \right) \right|_{\varepsilon=0} \\
&= \int_{\Omega} \left(\Phi'((R_{f,H}[u](y))^2) R_{f,H}[u](y) \right) \cdot \\
&\quad \cdot \left. \frac{d}{d\varepsilon} \left(f(y) - \int_{\Omega} (u(x) + \varepsilon v(x)) H(y, x) dx \right) \right|_{\varepsilon=0} dy \\
&= - \int_{\Omega} \left(\Phi'((R_{f,H}[u](y))^2) R_{f,H}[u](y) \right) \cdot \int_{\Omega} v(x) H(y, x) dx dy \\
&= - \int_{\Omega} \int_{\Omega} \left(\Phi'((R_{f,H}[u](y))^2) R_{f,H}[u](y) \right) \cdot H(y, x) dy \cdot v(x) dx . \quad (6.14)
\end{aligned}$$

Requiring that this equality holds for all admissible functions v , we can drop the integral and multiplication with v , which gives us the variational gradient

$$\frac{\delta}{\delta u} E_1[u](x) = - \int_{\Omega} \Phi'((R_{f,H}[u](y))^2) R_{f,H}[u](y) H(y, x) dy . \quad (6.15)$$

Notice that the arguments of H have switched their positions – the operator appearing here is the adjoint of the blur operator. In the case of a spatially invariant PSF h this corresponds to a convolution with \tilde{h} where $\tilde{h}(x) := h(-x)$ is the result of reflecting h at the origin.

Variational gradient of the regularisation term E_2 . The regularisation term E_2 is exactly of the same type as the energy functional (2.20) considered in 2.2.2 such that its variational gradient equals (2.25), i.e.,

$$\frac{\delta}{\delta u} E_2[u] = -\operatorname{div}\left(\Psi'(|\nabla u|^2) \nabla u\right). \quad (6.16)$$

Gradient descent. Using the variational gradients for both components of the energy functional (6.10), a gradient descent for the latter finally reads

$$\partial_t u = -\frac{\delta}{\delta u} E_1[u] - \alpha \frac{\delta}{\delta u} E_2[u] \quad (6.17)$$

or, by inserting (6.15) and (6.16),

$$\partial_t u = \int_{\Omega} \Phi'((R_{f,H}[u](y))^2) R_{f,H}[u](y) H(y, \cdot) dy + \alpha \operatorname{div}\left(\Psi'(|\nabla u|^2) \nabla u\right). \quad (6.18)$$

This integro-differential equation is the basic form of what we will refer to as *deblurring equation* in this part of the work.

The last summand on the right-hand side of (6.18) is a diffusion term. The first summand resembles a reaction term in a diffusion–reaction PDE. While a reaction term, however, is usually understood as a local interaction that acts as data-dependent source or sink, this term is not strictly local due to the integration. If the PSF has (locally) a support of a certain size, then the region of influence of the integral term in (6.18) has roughly speaking the double extent of that support. Having said this, we will for the sake of simple terminology nevertheless refer to the integral term of (6.18) as reaction term and also call the entire equation a diffusion–reaction equation.

6.3 Taxonomy of Energy Functionals and Corresponding Diffusion-Reaction Equations

So far, we have specified in our energy functional (6.10) neither the data term penaliser Φ nor the regularisation penaliser Ψ . We discuss relevant choices now.

6.3.1 Data Terms – Reaction Terms

Different data terms in the energy functional induce different reaction terms in the gradient descent equation.

6.3.1.1 Quadratic Data Term – Linear Reaction Term

The most widespread choice in variational deconvolution literature (as in most fields of variational image processing) is still the *quadratic Tikhonov penaliser* $\Phi(s^2) = s^2$, i.e.,

$$E_1[u] = \int_{\Omega} (R_{f,H}[u](x))^2 dx. \quad (6.19)$$

Via the variational gradient

$$\frac{\delta}{\delta u} E_1[u](x) = - \int_{\Omega} R_{f,H}[u](y) H(y, x) dy \quad (6.20)$$

it leads to linear reaction terms, thereby offering the advantage of utmost simplicity and applicability of the well-developed theory for linear operators and, after discretisation, linear systems of equations.

6.3.1.2 Robust Data Term – Nonlinear Reaction Term

In the context of discrete deconvolution models, robust data terms have already been proposed in 1995 [235], but it seems that this proposal attracted little attention at that time. They did not come into broader use until recently, when they were introduced in the variational deblurring framework by Bar et al. [15].

In the case of grey-value images, robustification means to replace the squared residual $(R_{f,H}[u])^2$ as penaliser by some nondecreasing function Φ of it such that $\Phi(s^2)$ grows slower than s^2 . The standard choice here is an L^1 penaliser, with $\Phi(s^2) = |s|$, or its regularised form $\Phi(s^2) = \sqrt{s^2 + \beta^2}$ with a small positive β . In the latter case the data term then takes the form

$$E_1[u] = \int_{\Omega} \sqrt{(R_{f,H}[u])^2 + \beta^2} dx, \quad (6.21)$$

corresponding to the variational gradient

$$\frac{\delta}{\delta u} E_1[u](x) = - \int_{\Omega} \left(\frac{R_{f,H}[u](y)}{\sqrt{(R_{f,H}[u](y))^2 + \beta^2}} \right) \cdot H(y, x) dy. \quad (6.22)$$

We do not consider in this work non-convex penaliser functions in the data term. In practice, already L^1 penalisers offer a considerable gain in robustness and thereby reconstruction quality, which is still not fully investigated in all its facets. On the other hand, non-convex penalisers introduce a difficult-to-control non-uniqueness into the model.³

³In the regularisation term, however, we will include non-convex penalisers, see Section 6.3.2.

6.3.1.3 Multi-Channel Case

Up to this point, all considerations of energy functionals and gradient descents in this chapter have been carried out for scalar-valued images. In order to devise deconvolution methods also for multi-channel images we need adequate formulations of data and regularisation terms with their respective gradient descents. We start here by considering data terms.

In the case of the quadratic penaliser $\Phi(s^2) = s^2$, an obvious choice is to use the sum of squares of residuals in the channels, i.e.,

$$E_1[u] = \frac{1}{2} \int_{\Omega} \sum_{k \in \Gamma} (R_{f_k, H}[u_k](x))^2 dx \quad (6.23)$$

where the index set Γ enumerates the image channels u_k of u .

This choice is obviously associated with the Euclidean norm in the multidimensional image range, which particularly for multi-channel data with geometric semantics (like vector or tensor fields) is advantageous from the modelling viewpoint as it ensures rotational invariance. Moreover, the reaction term resulting from the variational gradient

$$\frac{\delta}{\delta u_k} E_1[u](x) = - \int_{\Omega} R_{f_k, H}[u_k](y) H(y, x) dy \quad (6.24)$$

is linear and separable between image channels, which offers algorithmic advantages.

In fact, separability corresponds also to a specific model assumption: It assumes that noise is independent between image channels.

Separate versus joint robustification. The situation is more complicated in the case of a robust penaliser. Here, one can in general rescue only one of the two desirable properties, rotational invariance and separability.

Separability is achieved obviously if the penaliser function Φ is applied individually to each image channel, as in

$$E_1[u] = \frac{1}{2} \int_{\Omega} \sum_{k \in \Gamma} \Phi((R_{f_k, H}[u_k](x))^2) dx \quad (6.25)$$

which we will refer to as *separate robustification*. The variational gradient in this case takes the form

$$\frac{\delta}{\delta u_k} E_1[u](x) = - \int_{\Omega} \Phi'((R_{f_k, H}[u_k](y))^2) R_{f_k, H}[u_k](y) H(y, x) dy \quad (6.26)$$

for each channel $k \in \Gamma$.

In contrast, rotational invariance is achieved if Φ is applied to the Pythagorean sum of all image channels, i.e.,

$$E_1[u] = \frac{1}{2} \int_{\Omega} \Phi \left(\sum_{k \in I} (R_{f_k, H}[u_k](x))^2 \right) dx. \quad (6.27)$$

We will call this *joint robustification*. The corresponding variational gradient reads

$$\frac{\delta}{\delta u_k} E_1[u](x) = - \int_{\Omega} \Phi' \left(\sum_{l \in I} (R_{f_l, H}[u_l](y))^2 \right) R_{f_k, H}[u_k](y) H(y, x) dy, \quad (6.28)$$

displaying a coupling between channels via the argument of Φ' .

Note that both (6.25) and (6.27) are generalisations of (6.23). While intermediate choices are possible – providing in general neither separability nor rotational invariance, and bringing about algorithmic complication –, these are the two basic cases. Again, they can be associated with the assumptions on noise: Separate robustification assumes that noise is independent between channels, while joint robustification assumes a noise distribution that is strongly channel-coupled.⁴

As a rule of thumb, it is advisable to use joint robustification whenever channel-coupled noise is to be expected. This applies to most multi-channel imaging procedures, be it digital colour photography (where the optics is shared by all colour channels, and mostly all channels are recorded on one CCD chip) or DT-MRI (where an error in one diffusion-weighted measurement channel always influences several matrix channels). We will therefore use joint robustification in all multi-channel experiments presented in this work. In contrast, e.g. Bar et al. [13] advocate separate robustification without addressing the problematic assumption of noise independence between channels. For a discussion of joint versus separate robustification in a different context – optic flow – see also [45, p. 38].

6.3.2 Regularisers – Diffusion Terms

Correspondences between different energy functionals for regularisation and diffusion expressions arising as their gradient descent equations have been studied in Chapter 2. These considerations transfer directly to the regularisation term $E_2[u]$ of our deconvolution energy functional. We can therefore restrict ourselves to repeating only the most essential points. Building on the framework described in Chapter 2 that embeds also smoothly those diffusion processes for which no variational gradient descent representation is known, we are led to investigate also the use of the latter

⁴It is worth noticing that the one situation in which joint and separate robustification coincide corresponds to quadratic penalisation $\Psi(s^2) = s^2$. Statistical considerations link this case to Gaussian noise which due to its separability can indeed be considered equally as channelwise (independent) or channel-coupled noise.

diffusion terms in the deconvolution context. We will speak in this case of *variationally motivated* deblurring equations.

6.3.2.1 Quadratic Tikhonov-Type (H^1) Regulariser

The simplest case, the constant diffusivity $g(s^2) = 1$ which corresponds to Tikhonov regularisation $\Psi(s^2) = s^2$, leads to a linear diffusion term in the deblurring equation. From the view point of image processing, it often tends towards an over-smoothed deblurring result because high gradients at edges of the reconstructed image are penalised over-proportionally.

Moreover, the resulting deblurring equation is very close to a linear process (it is even linear if also a quadratic data term is used). It comes as no surprise that it also suffers from the same kind of artifacts as described above for linear deconvolution methods.

6.3.2.2 Non-Quadratic Regularisation and Isotropic Nonlinear Diffusion

A non-quadratic penaliser $\Psi(|\nabla u|^2)$, where $\Psi(s^2)$ typically grows slower than s^2 , gives rise to a nonlinear isotropic diffusion term $\operatorname{div}(g(|\nabla u|^2)\nabla u)$, where $g \equiv \Psi'$. Depending on the choice of Ψ (thus, g) one has either an edge-preserving or even an edge-enhancing regulariser. We name again the two main representatives.

Total variation. The singular diffusivity function $g(s^2) = 1/|s|$ is associated with the L^1 penaliser $\Psi(s^2) = |s|$. Mostly in its regularised form $g(s^2) = 1/\sqrt{s^2 + \varepsilon^2}$, it is a popular choice in the variational deconvolution literature, see e.g. [59, 138, 14]. This is the case on one hand because it enforces piecewise constant results and therefore preserves edges in the image. On the other hand, $\Psi(s^2)$ is still convex w.r.t. s in this case, therefore providing uniqueness of assertions about minimisers that are not guaranteed to hold with non-convex penalisers.

Perona–Malik. A drawback of TV diffusivity is that it does not allow an enhancement of edges. Using the Perona–Malik diffusivity $g(s^2) = (1 + s^2/\lambda^2)^{-1}$ associated to the nonconvex regulariser $\Psi(s^2) = \lambda^2 \ln(1 + s^2/\lambda^2)$, see [161, 206], makes this possible. However, this happens at the cost that the smoothness energy $\Psi(|\nabla u|^2)$ is no longer convex. It has therefore to be expected that depending on the initial conditions different solutions are obtained, making the choice of initial conditions a relevant issue.

Pre-smoothing. To reduce the noise sensitivity of isotropic Perona–Malik diffusion (see [205, 206]) it can be stabilised by using a Gaussian-smoothed gradient ∇u_σ

in the diffusivity argument, turning the diffusion expression into $\operatorname{div}(g(|\nabla u_\sigma|^2)\nabla u)$. This stabilised Perona–Malik diffusivity can easily be plugged into our deblurring equations, which in this case ceases to be a gradient descent for an energy functional. Experiments indicate, however, that in the deblurring context Perona–Malik diffusion with pre-smoothing bears no clear advantages over its counterpart without pre-smoothing.

6.3.2.3 Anisotropic Nonlinear Diffusion

In analogy to the proceeding in 2.3 one is led to substituting the isotropic diffusivity g with an anisotropic diffusion tensor $D(\nabla u_\sigma)$ in order to achieve an improved reconstruction of edges. In our model, we use $D(\nabla u_\sigma) = g(\nabla u_\sigma \nabla u_\sigma^T)$ where the Perona–Malik diffusivity g is applied to the symmetric matrix $\nabla u_\sigma \nabla u_\sigma^T$, as usual, by letting g act on the eigenvalues and leaving the eigenvectors unchanged [206].

The resulting *anisotropic deblurring equation*

$$\partial_t u = \int_{\Omega} \Phi'((R_{f,H}[u])^2) R_{f,H}[u] H(y, \cdot) dy + \alpha \operatorname{div}(D(\nabla u_\sigma) \nabla u) \quad (6.29)$$

contains a smoothed gradient in the argument of the diffusion tensor. This smoothing is inevitable in order to have true anisotropy. As a consequence, (6.29) is not known so far to be the gradient descent for an energy. It is therefore appropriate to speak of a variationally motivated deblurring method in this case.

6.3.2.4 Multi-Channel Case

As the regularisers discussed here lead to diffusion expressions that stand in full analogy to those considered in Chapter 2, the generalisation to multi-channel images is straightforward, too.

While linear diffusion (see 6.3.2.1) is simply applied in each image channel separately, the proceeding in the case of isotropic (see 6.3.2.2) and anisotropic nonlinear diffusion (see 6.3.2.3) follows that in 2.2.5 and 2.3.2, respectively: The divergence expression is computed in each image channel separately but the diffusivity $g(\cdot)$ or diffusion tensor $D(\cdot)$ is uniform across all channels.

In the same way as in Chapter 2, this treatment of multi-channel images is linked to energy functionals that combine gradient information from all channels into one energy term.

6.4 Special Model Aspects

6.4.1 Outliers and Violation of Model Assumptions

The noise model underlying robust data terms is in fact a random process that acts independently and identically distributed on all pixels. The difference to Gaussian noise that gives rise to quadratic data terms is one has here a noise distribution with “thick tails”. Nevertheless, the robustness achieved by the corresponding data terms goes considerably beyond this assumption.

Following the principle to consider influences on which insufficient information is present and that cannot be controlled for as random, one can include in the noise many kinds of perturbations other than pixel noise. In other words, any violation of the blur model that is encoded in the residual formula can be considered as a sort of noise.

However, the resulting concept of noise differs from the classical assumption formulated above in that in the noise now there will be significant autocorrelation and spatial variation. Nevertheless, it turns out that indeed robust data terms can cope with several types of model violations, apparently because the assumption of thick-tail pixel noise is already more extreme than those. We will now name one situation in which this robustness is useful. Another example is discussed in 6.4.2.3.

Imperfect knowledge on PSF. One type of model violations in a non-blind deconvolution model that implies a strong perturbation with spatial correlation is imprecision in the estimated PSF. Even small imprecision here often deteriorates considerably deblurring results with non-robust techniques. In contrast, our robust deconvolution model performs favourable even in the presence of visible deviations from the PSF estimate. We will demonstrate this experimentally later.

6.4.2 Boundary Treatment

For solving the deblurring equations (6.18) (or (6.29)), suitable boundary conditions must be specified. Due to the integro-differential character of our deblurring equations, the problem of boundary conditions is even worse than for differential equations: Since the convolution expression in the data term smears out boundary information instantaneously to locations in the interior of the image domain, there is a more or less extended region in the interior of the image domain on which the boundary conditions take direct effect. Moreover, boundary conditions can involve dramatic violations of the blur model encoded in the data term. The resulting large

penalties in the data term can dominate the behaviour of the solution in a region that sometimes extends deeply into the interior of the image domain.

In the following, we sketch a few approaches to this problem, with emphasis on their practical usefulness.

6.4.2.1 Image Extension

A class of more or less simple approaches attempts at extrapolating the given blurred image beyond its boundaries in some way such that the extrapolated data are approximately consistent with the blur assumption, compare Figures 6.2 and 6.3(a).

Neumann boundary conditions. In partial differential equations like e.g. diffusion equations one possible choice is given by Neumann boundary conditions which require that the image derivative perpendicular to the boundary vanishes on the boundary. In the case of a rectangular image domain this is equivalent to continuing the image beyond each boundary by a mirrored version of itself (Figure 6.2(c)). They are therefore also referred to as reflecting boundary conditions. Neumann boundary conditions are often an excellent choice in diffusion-based image processing methods, as they guarantee conservation properties along with a continuous extension of the image at its boundary.

Unfortunately, reflecting boundary conditions are badly suited to the deconvolution context: The reflected extension of the image is approximately the blurred version of the reflected sharp image with the *reflected* point spread function. This is acceptable only in the case of a symmetric point spread function, such as a Gaussian, a defocussing disc or a motion blur parallel or perpendicular to the boundary. With non-symmetric PSF, it constitutes a substantial violation of spatially invariant or spatially continuous blur models.

Moreover, reflecting boundary conditions will in most cases violate spatially continuous blur models in the immediate neighbourhood of the boundary whatever the PSF is (even if it is symmetric). This is due to the cut-off of the blurring process at the image boundary.

Both mentioned effects cause strongly visible artifacts.

Periodic boundary conditions. Periodic boundary conditions (Figure 6.2(d)) for a rectangular domain lead instead to a wrap-around of image information between opposite boundaries. In this case, the image extension is in accordance with a space-invariant blur model. Also, periodic boundary conditions are in natural correspondence to the Fourier transform which is used in the Wiener filter and can also be



Figure 6.2: *Left to right:* (a) *Styrian Hog* photograph, 240×320 pixels. – (b) Blurred with linear motion blur (40 pixels horizontal, 20 pixels vertical, compare Figure 6.1(c)). (a) and (b) were clipped from a larger image, such that the information transport across the boundary is correctly simulated. – (c) Blurred image extended by mirroring, corresponding to reflecting boundary conditions. The blur assumption is severely violated in the amended regions. No discontinuities are introduced. – (d) Periodic image extension. The blur assumption is fulfilled in the interior of the amended regions. It is violated at the boundaries themselves, where also discontinuities are introduced. – (e) Zero-padding extension. Violation of blur assumptions and introduction of discontinuities are analog to (d). – Images (c), (d): D. Theis.

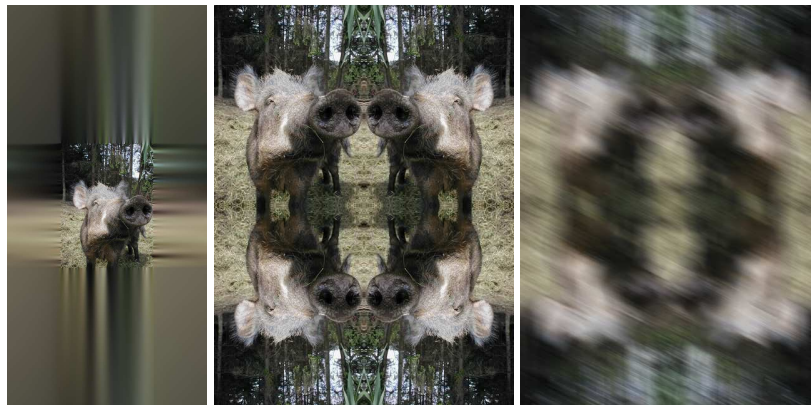


Figure 6.3: *Left (a):* Continuous and periodic extension of an image by weighted averages of Gaussian smoothed copies of the first and last rows and columns. For greater clarity, the extension is shown here for the sharp original *Styrian Hog* image. The extended image size has been set to 512×1024 pixels for convenient FFT use. – *Middle (b):* Continuous periodic image consisting of four mirrored copies of the sharp *Styrian Hog* image for the special test setting. – *Right (c):* Image (b) blurred by linear motion blur (40 pixels horizontal, 20 pixels vertical) with periodic boundary treatment. This type of blurred image is used in the special test setting. – Modified from images generated by D. Theis.

useful in computing the convolutions within a numerical realisation of our deblurring integro-differential equations.

While the first drawback of periodic boundary conditions is thereby elegantly avoided, the violation of the blur model in the boundary region itself remains. This problem gets even worse since unlike reflecting boundary conditions, periodic ones introduce discontinuities at the boundaries. As a result, not only the wrap-around of image information between opposing boundaries is observed which is always typical for periodic boundary conditions, but also strong oscillatory (“ringing”) artifacts parallel to boundaries are observed.

Nevertheless, due to the above-mentioned properties, periodic boundary conditions are an option especially for space-invariant PSFs.

Zero padding. A third possibility is to surround the image by a zone of zero (or other constant) grey values (Figure 6.2(e)) and apply e.g. periodic boundary conditions to the so extended image domain. In this case, the benefits of periodic boundary conditions are retained while the wrap-around artifacts are eliminated. Moreover, the constant image extension is able to fit any blur model. However, also the zero-padding procedure introduces discontinuities along the boundaries that imply similar ringing artifacts as before.

Other extensions. It requires considerable effort to design boundary regimes that really eliminate the sources of artifacts that have been described for the methods so far. A simple, more pragmatic way is to design more sophisticated image extensions that guarantee continuity at the boundaries and an approximate fit to the blur model. If periodic boundary conditions are to be used for the extended image, one can in particular also secure continuity between opposing boundaries.

In [193] a method has been proposed which to this end continues the image by weighted averages of Gaussian blurred boundary rows and columns, see Figure 6.3(a). Here, the rectangular image domain is embedded into a larger periodic domain (a torus). To fill the uncovered parts of the torus, one continues the image by copies of its boundary rows and columns which are increasingly blurred by 1D Gaussian convolution, and takes weighted averages between opposing boundary rows or columns. In [15] an image version blurred with the adjoint of the (spatially invariant) point spread function was used to generate the extension. These methods often lead to a reasonable reduction, though not elimination, of artifacts. Their simplicity makes them a reasonable choice in many cases.

6.4.2.2 A Special Test Setting

In the case of synthetically generated test cases, it is possible to eliminate boundary artifacts in order to measure the deblurring quality by a special test setting. This has been described in [193, 222].

We assume here a spatially invariant but possibly irregularly shaped point spread function. We start by extending the *sharp* test image via horizontal and vertical reflection to quadruple size, see Figure 6.3(b). Periodic extension of this larger image is equivalent to reflecting extension of the original image, thus not involving discontinuities. Now the large image is blurred in a periodic setting (i.e., with the left boundary wrapping over into the right one etc.) with the given PSF. While the resulting image, see Figure 6.3(c), has lost the symmetry of the original larger image, it perfectly fulfils the blur model assumption throughout the periodic image domain, without any deviation at boundaries. In deblurring this image, one uses again periodic boundary conditions.

While this approach cannot be used in real deblurring applications where the blurring process is not subject to our control, its advantage is that it admits a measurement of the deblurring quality, e.g. in terms of signal-to-noise ratio, without including discontinuity and boundary artifacts which would dominate the total result otherwise.

6.4.2.3 Modelling via the Point-Spread Function

The formulation of our deblurring model with a space-variant point-spread function opens another way to deal with the problem of information transport across the image boundary.

For this purpose, we modify the point-spread function in such a way that it is defined on $\Omega \times \Omega$ only, and that

$$\int_{\Omega} H(x, y) dx = 1 \quad (6.30)$$

for each $y \in \Omega$. The modification is to some extent arbitrary but it can be restricted to a narrow region along the image boundary. As a consequence, H models a blur process in which the grey-value contribution of each location x within the image domain remains within the image.

The deviation from this assumption is considered as noise. This can be seen as another instance of the idea brought up in 6.4.1 to include violations of model assumptions into the noise. Indeed, influences from beyond the image boundary cannot be controlled for in the blur model, such that it is reasonable for them to be considered as random. However, once more simple noise models will no longer be adequate, as the boundary effects introduce significant autocorrelation into the noise. Therefore, the

treatment of boundary effects as noise can work only if the deblurring model is able to cope with such types of noise. Fortunately, robust data terms again match this need. It turns out that they perform fairly well also in this situation.

6.4.2.4 Boundary Treatment by Inpainting

A more radical way to cope with boundary effects in image deblurring has been proposed in [60]. The authors integrate (blind) deconvolution with image inpainting on a variational basis. The inpainted region can as well be chosen as the exterior of the image domain, which is therefore extrapolated in a way that matches the information found within the blurred image more perfectly than a model with fixed boundary conditions can.

While it is to be expected that reconstruction results near the boundary are better with this approach than with any other method discussed here, we give preference here to the method from 6.4.2.3, which performs reasonably good in our examples. The combination of deconvolution with other image processing tasks like inpainting is beyond the scope of the present work.

6.4.3 Choice of Regularisation Weight

Whenever a regularisation approach is used, choosing the diffusion weight appropriately is an important issue. The deblurring problem with its severe ill-posedness adds to this difficulty since, as we will see, it assigns a double purpose to the regularisation. We will discuss this now in the context of variational deconvolution processes.

6.4.3.1 The Double Role of Regularisation

As in many regularisation models, regularisation also in variational deconvolution serves the purpose of suppressing noise. Regularisers can be adapted to noise models, and measures of noise intensity can be linked to suitable values of the regularisation weights. In general, for stronger noise larger values of α need to be chosen.

However, there is another reason why regularisation is needed in variational deconvolution, which is related to a deficiency in the data terms used. In absence of noise, the non-regularised energy consisting only of the data term is minimised by the correct solution. One could therefore consider to set α to zero, thereby simplifying the diffusion-reaction equation (6.18) to a fixed-point equation without the diffusion part.

Unfortunately, the solution of the minimisation problem without regularisation can be highly non-unique. The reason is that the data term is insensitive to such pertur-

bations which are annihilated by the blurring with the point-spread function H . For example, if $H(x, y) = h(x - y)$ is spatially invariant, each location in the Fourier domain h corresponds to a spatial frequency which is extinguished by the blur. Perturbations constructed from these frequencies are eigenfunctions for the blur operator with eigenvalue zero, and are thus not corrected by the data term. The energy to be minimised is still convex, but not strictly convex, in this situation.

Even in the case of a PSF without such zero-eigenvalue eigenfunctions, e.g., space-invariant Gaussian blur, some frequencies are strongly attenuated by the blur, making the solution of the minimisation problem highly unstable. As a consequence, regularisation with a non-zero α is needed to remove or reduce the non-uniqueness. With a convex regulariser, strict convexity of the energy functional can be ensured under suitable conditions. With a non-convex regulariser of Perona-Malik type, this needs not to be the case; in practice, it still provides a disambiguation effect.

While this latter discussion seems to favour a larger α even in case of low noise, this would be a premature conclusion. Assume, namely, that the energy is made convex by a suitable regulariser such that a unique solution exists. Even if the regularisation weight is small, it is practically only the regularisation term which chooses the solution among those which cannot be discriminated by the data term. As a consequence, even a small regularisation weight can drive the solution far away from the true unblurred image.

Experimentally one observes that when using the blurred image as initialisation with small diffusion weight, similar oscillatory artifacts as for linear deblurring methods evolve, like over- and undershoots near edges and shadows of high-contrast objects. Large diffusion weights, on the other hand, induce an over-smoothing and loss of small-scale details in the image.

6.4.3.2 Continuation Strategy

In the conflict between accurate restoration of image details (which is better achieved for small α) and suppression of artifacts (which requires a large α) it is useful to notice the dependence of the solution on initial conditions. In case of nonconvex regularisation there can exist multiple minima which can be reached from different initial data. Even with convex regularisers, the non-strict convexity of the data term can lead to considerable numerical errors in the computed minimiser when the regularisation weight is low, which depend similarly on the initial conditions.

This influence of initial conditions opens a way how to a certain extent the advantages of large and small regularisation weights can be combined in practice in a *continuation strategy* [30, 58]: Instead of considering only one energy functional with a fixed amount of regularisation, a cascade of functionals is taken into account. Starting with

a rather large amount of regularisation, this amount is reduced from one member of the cascade to the next. The level to which the regularisation weight can be reduced is mainly determined by the first mentioned purpose of the regularisation, namely noise suppression. For images with very low noise, the final α can be close to zero – experimentally one can sometimes even use $\alpha = 0$ which means that the inevitable numerical dissipation provides sufficient regularisation.

In such a cascade of minimisations, the first steps prefer smooth solutions and therefore may yield good initialisations close to the smoothest optimum of the next version with less regularisation. On the other hand, in later steps more and more the data term dominates which models the blur model and favours sharper results. This way, one finally approximates an optimum of the functional without regularisation, yet choosing a specific optimum that is smooth. Though this optimum will not be exactly the original image – in particular, it tends to be slightly over-smoothed –, it will often be superior to solutions obtained without continuation strategy, provided the smoothness assumptions encoded in the regularisation term are adequate for the type of image being processed.

6.5 Numerical Aspects

The focus on the work on variational deblurring presented here is in the development of models. In this section we present briefly explicit finite-difference schemes that we use in our experiments to evaluate the deblurring equations (6.18), (6.29) and further variants. We stress that this numerical realisation has primarily a proof-of-concept character. Improvements are possible and necessary, and future work will be devoted to them.

For our considerations in this section, we will adopt the same convention as introduced in 4.2.3: If $z = z(x, t)$ is an expression on a continuous domain, we will denote by $[z]_{ij}^K$ a discretisation of it at location (i, j) in time step K .

To start with, the time derivative on the left-hand side of (6.18), (6.29) etc. will always be discretised by a forward difference

$$[u_t]_{ij}^K = \frac{u_{ij}^{K+1} - u_{ij}^K}{\tau} \quad (6.31)$$

where τ denotes the time-step size.

The right-hand side decomposes into the reaction and diffusion terms which we will now discuss separately.

6.5.1 Discretisation of the Reaction Term

The computationally most expensive task in the numerical evaluation of the deblurring equations (6.18), (6.29) and further variants is the computation of the reaction term. It contains convolution-type integrals that need to be computed in each single time step. Depending on the specific situation different realisations are used.

Convolution via the Fourier domain. This is applicable if a spatially invariant PSF is used. The computational cost of the Fourier transforms pays off if the support of the PSF h is large and if the image dimensions enable the use of an efficient fast Fourier transform.

In the case of quadratic data terms, $|\hat{b}|^2 = \widehat{\tilde{h} * h}$ is precomputed⁵, leaving two Fourier transforms (one forward and one backward transform) of the image per time step.

With robust data terms, \hat{b} is precomputed, and four Fourier transforms of the image (two forward and two backward transforms) are necessary since Ψ' is computed in the spatial domain between the two convolutions.

Convolution in the spatial domain. Especially for point-spread functions with small support direct convolution in the spatial domain is often faster than the Fourier transforms. Again, for non-robust data terms precomputing $\tilde{h} * h$ cuts the computational cost nearly by half.

Spatially variant PSF. For spatially variant point-spread functions, we always evaluate the reaction term in the spatial domain. Since the memory demand of a full spatially variant PSF precludes its full storage, it is recovered during each integral computation by interpolation or by parametric representations from a few data, depending on the model used for its specification. This adds to the computational cost of the integration itself, increasing it up to four times.

For a more detailed analysis, let us concentrate on the evaluation via the spatial domain. Discretising integrals into sums and replacing all quantities by their discrete counterparts from time step K , we obtain as our discretisation of the reaction term in pixel (i, j) the expression

$$P_{i,j}^K := \sum_{(k,l) \in \mathcal{H}^*(i,j)} [R_{f,H}]_{k,l}^K \Phi'([R_{f,H}]_{k,l}^K)^2 H_{(k,l)(i,j)} \quad (6.32)$$

with the discretised residual

$$[R_{f,H}[u]]_{k,l}^K = f_{kl} - \sum_{(s,t) \in \mathcal{H}(k,l)} H_{(k,l)(s,t)} u_{s,t}^K. \quad (6.33)$$

⁵Remember $\tilde{h}(x) := h(-x)$, cf. page 223

Here we have used H also to denote the matrix that arises from discretising the point-spread function H . Then $H_{(k,l)(i,j)}$ is the matrix entry corresponding to $H(x,y)$ at location (k,l) for x and (i,j) for y , i.e., it quantifies the influence of pixel (i,j) on (k,l) in the blur operation.

By $\mathcal{H}(i,j)$ we denote a neighbourhood of (i,j) which contains the support of $H_{(i,j)(\cdot)}$, by $\mathcal{H}^*(i,j)$ a neighbourhood containing the support of $H_{(\cdot)(i,j)}$. Formally, one can of course use the entire image in both cases.

Inserting (6.33) into (6.32) we arrive at

$$\begin{aligned} P_{i,j}^K &= \sum_{(k,l) \in \mathcal{H}^*(i,j)} \varphi(\left(\left[\mathcal{R}_{f,H}\right]_{k,l}^K\right)^2) H_{(k,l)(i,j)} f_{k,l} \\ &\quad - \sum_{(k,l) \in \mathcal{H}^*(i,j)} \sum_{(s,t) \in \mathcal{H}(k,l)} \varphi(\left(\left[\mathcal{R}_{f,H}\right]_{k,l}^K\right)^2) H_{(k,l)(i,j)} H_{(k,l)(s,t)} u_{s,t}^K. \end{aligned} \quad (6.34)$$

By setting

$$F_{i,j}^K := \sum_{(k,l) \in \mathcal{H}^*(i,j)} \Phi'(\left(\left[\mathcal{R}_{f,H}\right]_{k,l}^K\right)^2) H_{(k,l)(i,j)} f_{k,l} \quad (6.35)$$

and

$$B_{(i,j)(s,t)}^K := \sum_{(k,l)} \Phi'(\left(\left[\mathcal{R}_{f,H}\right]_{k,l}^K\right)^2) H_{(k,l)(i,j)} H_{(k,l)(s,t)} \quad (6.36)$$

we rewrite (6.34) into

$$P^K = F^K - B^K u^K. \quad (6.37)$$

For *non-robust* data terms, one has of course $\varphi \equiv 1$, thus $F^K = F$ and $B^K = B$ are independent on the time step index K .

6.5.2 Discretisation of the Diffusion Term

As for the left-hand side, we use finite difference discretisations in the diffusion term. A general expression for the discretised diffusion term in pixel (i,j) at time step K is given by

$$Q_{i,j}^K := \sum_{(k,l) \in \mathcal{N}(i,j)} \omega_{(k,l)(i,j)} g_{(k,l)(i,j)}^K (u_{k,l}^K - u_{i,j}^K), \quad (6.38)$$

where $\mathcal{N}(i,j)$ denotes a suitable neighbourhood of pixel (i,j) , and $\omega_{(k,l)(i,j)}$ is the weight of pixel (k,l) in the neighbourhood of (i,j) . Finally, $g_{(k,l)(i,j)}^K$ denotes the discrete diffusivity between pixels (i,j) and (k,l) in time step K . In the isotropic case, this is a discretisation of $\Psi'(|\nabla u|^2)$, while in the anisotropic case it is obtained in a more complex way from the diffusion tensor D .

While the weights ω are quantities inherent to the discretisation, and reflect geometric properties of the grid, the diffusivities g depend on the image u , which also

entails the appearance of a time-step index on them. Both weights and diffusivities are required to be symmetric,

$$\omega_{(i,j)(k,l)} = \omega_{(k,l)(i,j)}, \quad g_{(i,j)(k,l)}^K = g_{(k,l)(i,j)}^K, \quad (6.39)$$

in order to ensure the conservation property of the diffusion process. For regular grids, the weights will also be translation invariant,

$$\omega_{(i+r,j+s)(k+r,l+s)} = \omega_{(i,j)(k,l)}. \quad (6.40)$$

The expression (6.38) is fairly general. It can model standard explicit schemes as well as, e.g., four-pixel explicit schemes. We concentrate here on the case of nonlinear isotropic diffusion terms as in (6.18).

Standard discretisation. For a standard explicit discretisation with spatial mesh sizes Δx in x direction, Δy in y direction we have

$$\begin{aligned} \mathcal{N}(i,j) &= \{(i-1,j), (i+1,j), (i,j-1), (i,j+1)\} \\ \omega_{(i,j)(i\pm 1,j)} &= \frac{1}{\Delta x} \\ \omega_{(i,j)(i,j\pm 1)} &= \frac{1}{\Delta y}, \end{aligned} \quad (6.41)$$

and the diffusivity is given as

$$g_{(i,j)(k,l)}^K = \frac{1}{2}(g_{i,j}^K + g_{k,l}^K) \quad (6.42)$$

with $g_{s,t}^K := \Psi'([\|\nabla u\|_{s,t}^2]^K)$, i.e., an average of discretisations of $\Psi'(|\nabla u|^2)$ at neighbouring pixel positions. Alternatively, one could consider discretising diffusivities directly in the locations half-way between neighbouring pixels,

$$g_{(i,j)(k,l)}^K = g_{(i+k)/2,(j+l)/2}^K. \quad (6.43)$$

Four-pixel discretisation. Assuming equal mesh sizes $\Delta x = \Delta y$, another possibility is an explicit weighted four-pixel discretisation in the sense of (4.65). With the

discretisation parameter $\gamma \in [0, 1]$, we have then⁶

$$\begin{aligned}
 \mathcal{N}(i, j) &= \{(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), \\
 &\quad (i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\}, \\
 \varpi_{(i,j)(i\pm 1, j\pm 1)} &= \frac{1-\gamma}{2\Delta x}, \\
 \varpi_{(i,j)(i\pm 1, j)} &= \varpi_{(i,j)(i, j\pm 1)} = \frac{\gamma}{\Delta x}, \\
 \mathfrak{g}_{(i\pm 1, j\pm 1)(i, j)} &= \mathfrak{g}_{i\pm 1/2, j\pm 1/2}, \\
 \mathfrak{g}_{(i\pm 1, j)(i, j)} &= \frac{1}{2}(\mathfrak{g}_{i\pm 1/2, j-1/2} + \mathfrak{g}_{i\pm 1/2, j+1/2}), \\
 \mathfrak{g}_{(i, j\pm 1)(i, j)} &= \frac{1}{2}(\mathfrak{g}_{i-1/2, j\pm 1/2} + \mathfrak{g}_{i+1/2, j\pm 1/2}).
 \end{aligned} \tag{6.44}$$

In this case, the discretisations $\mathfrak{g}_{i+1/2, j+1/2} := \Psi'([\|\nabla u\|^2]_{i+1/2, j+1/2})$ are computed in the manner of (4.72).

Matrix-vector product notation. For later analysis, we point out that in any case the right-hand sides of (6.38) for all pixels can be combined in a matrix-vector product notation

$$Q^K = A^K u^K \tag{6.45}$$

where A^K is a $N \times N$ matrix, N being the number of pixels in the image, and u^K is a column vector with the grey-values of all N pixels as its entries. The matrix A^K contains all weight and diffusivity contributions of (6.38).

Practical remarks. In the experiments in this and the following chapter both discretisations described above have been used. Experimentally, the influence of the discretisation of the diffusion term on the overall sharpness of the deblurred image turns out negligible. Results obtained with both types of explicit discretisations look practically the same.

In principle, one could even use the time step of our locally semi-analytic or analytic schemes to compute the diffusion contribution of (6.18). However, in the light of the stability analysis that will be given below, the absolute stability provided by the L(S)AS is pointless here. We have therefore not made use of this possibility.

With the implementations outlined here, computation times for the images shown here range from a few minutes to several hours on standard PCs.

⁶The discretisation parameter α from (4.65) had to be renamed to γ in the present chapter to avoid confusion with the regularisation weight α .

6.5.3 Stability Considerations

Combining the discretisations described in detail above, we have the explicit scheme

$$u^{K+1} = (I + \tau\alpha A^K - \tau B^K)u^K + \tau F^K \quad (6.46)$$

for the diffusion-reaction equation (6.18).

Concerning stability, we remark first that due to the character of the deblurring problem and the corresponding reaction term the scheme does not obey a maximum-minimum principle. Nevertheless, both in L^∞ and L^2 sense the scheme is stable for sufficiently small τ w.r.t. perturbations in the initial data u^0 or given image f .

A full stability analysis of (6.46) is intricate, as it requires to take into account the fact that also the matrices A^K, B^K and the vector F^K depend on u^K . We leave this detailed analysis for future work. Here we give instead a simplified analysis that neglects the dependency of A^K, B^K, F^K on u^K . In strict sense, the analysis is therefore only applicable to the case of linear diffusion and non-robust data term, in which these dependencies do not take place. Moreover, we consider only perturbations in the initial data u^0 but not those in f .

The equations we are dealing with are of reaction-diffusion type, and require a stability concept that differs somewhat from that for pure diffusion equations. We will consider the scheme (6.46) stable in L^2 norm w.r.t. perturbations of u^0 if for any fixed set of matrices A^K, B^K, F^K , and initial data u^0 there exists a constant M that does not depend on K , such that for sufficiently small $\|\tilde{u}^0 - u^0\|$ one has $\|\tilde{u}^K - u^K\| \leq M \cdot \|\tilde{u}^0 - u^0\|$ for all K .

In this case, a necessary condition for L^2 stability is that the matrix $C^K := I + \tau\alpha A^K - \tau B^K$ has only eigenvalues of absolute value not exceeding 1.

Let us therefore analyse the eigenvalues of C^K .

Consider A^K first. Because of our symmetry assumption on g , A^K is a symmetric matrix, featuring real eigenvalues. Remembering our assumption that $g(0) = 1$ and thus $0 \leq g(z^2) \leq 1$ for all z , we see that A^K has zero row sums and main diagonal entries from $[-4, 0]$. According to Gershgorin's Theorem it follows that for all eigenvalues λ of A^K the inequality

$$-8 \leq \lambda \leq 0 \quad (6.47)$$

holds.⁷

For B^K , we notice that (6.36) can be rewritten as

$$B_{(i,j)(s,t)}^K = \sum_{(k,l)} H_{(k,l)(i,j)} \varphi_{k,l}^K H_{(k,l)(s,t)}, \quad (6.48)$$

⁷If the four-pixel explicit discretisation (6.44) is used, the main diagonal entries are even constrained to $[-2(1+\gamma), 0]$, yielding the estimate $-4(1+\gamma) \leq \lambda \leq 0$.

where $\varphi_{k,l}^K := \Phi'([R_{f,H}]_{k,l})^2$, thus

$$B^K = H^T \text{diag}(\varphi_{k,l}^K)H \quad (6.49)$$

is symmetric and has therefore real eigenvalues. Moreover, as $\Phi'(z^2)$ is assumed to be nonnegative for all z , we can conclude further that

$$B^K = (L^K)^T(L^K) \quad (6.50)$$

where

$$L^K := \text{diag}\left(\sqrt{\varphi_{kl}^K}\right)H \quad (6.51)$$

Consequently, the eigenvalues μ_i of B can be computed from the eigenvalues v_i of L^K by

$$\mu_i = \bar{v}_i v_i = |v_i|^2 \quad (6.52)$$

(remember \bar{v}_i are the eigenvalues of $(L^K)^T$) ensuring B is positive semidefinite.

Further, as the matrix H describes the PSF of a blur, it is nonnegative, and the contributions of every pixel (i, j) of the unblurred image to the blurred image sum up to 1. That is to say, H has unit column sums.⁸ Since $0 < \Phi'(z^2) \leq 1$, the entries of L^K are nonnegative, too, and also its column sums are not greater than 1. By Gershgorin's Theorem we can therefore conclude that the eigenvalues v_i of L^K are located within the unit circle in the complex plane. By virtue of (6.52) we have then

$$0 \leq \mu_i \leq 1 \quad (6.53)$$

for the eigenvalues of B^K .

Putting together the bounds on the eigenvalues of A^K and B^K that we have found, we see that the eigenvalues of C^K (which are real, as C^K is diagonal, too) are confined to the interval $[1 - (1 + 8\alpha)\tau, 1]$.

Therefore the necessary stability condition is fulfilled if

$$\tau \leq \frac{2}{1 + 8\alpha}. \quad (6.54)$$

We stress again that this is not a sufficient condition for stability. However, the bound (6.54) is in good agreement with experimental observations. In practice, one does not immediately observe numerical instabilities if slightly larger time steps are used than admitted by (6.54). The reason is that for realistic point-spread functions the eigenvalues of the reaction term will often stay far away from the bound (6.53).

Because of the smallness of typical regularisation parameters α , the influence of the diffusion term on the stability condition is subordinate to that of the reaction term. This makes also clear why using the full L(S)AS construction for the diffusion term is of little use here: Even with an absolute stable discretisation of the diffusion term, the reaction term would still dictate $\tau \leq 2$.

⁸More precisely, the column sums of H are not greater than 1 if we take into account that pixels near the image boundary can be smeared out across the image boundary.



Figure 6.4: *Left to right: (a) Styrian Hog photograph, 240×320 pixels. – (b) Blurred with banana-shaped kernel from Figure 6.1(e). – (c) Deblurred by diffusion–reaction method with Perona–Malik diffusivity, $\lambda = 5$, $\alpha = 0.001$, 1000 iterations. – (d) Photograph blurred with discontinuous kernel. – (e) Deblurred by diffusion–reaction method with Perona–Malik diffusivity, $\lambda = 1$; the continuation strategy was used with two steps for the diffusion weight: 2400 iterations with $\alpha = 0.01$ followed by 2400 iterations with $\alpha = 0$. – From [222]. Deconvolved images by D. Theis.*

6.6 Experiments

6.6.1 Deblurring of Synthetically Blurred Images

To illustrate and validate the methods described in the preceding sections, we first show experimental results obtained with two test images and two different spatially invariant PSFs, Figure 6.1.

One of them (Figure 6.1(e)) is a banana-shaped blob with irregularly distributed intensity. This comes close to the blurring of photographs taken with bad illumination and moving camera and objects. The other PSF (Figure 6.1(d)) is discontinuous; it is assembled from two line-shaped parts which are similar to motion blurs. It has been selected as an example of a very challenging spatially invariant PSF.

The first test image used in Figs. 6.4 and 6.5 is a colour photograph with many small-scale details. The second test image used in Figs. 6.6 and 6.7 is a grey-value image of three print letters. It differs from the first image by its composition of fairly homogeneous regions.

Synthetically blurred *Styrian Hog* image. In Figure 6.4 we blur the first test image with both kernels and restore it by diffusion–reaction deblurring with Perona–Malik diffusivity. For the discontinuous kernel, we also use the continuation strategy in a simple form with one positive α followed by a fixed-point iteration with $\alpha = 0$. Excellent deblurring quality is achieved for the banana kernel (despite its irregularity) while for the discontinuous kernel some shadow-like boundary artifacts are observed.



Figure 6.5: Deblurring of a detail of the *Styrian Hog* photograph from Figure 6.4 with different boundary treatment. *Top, left to right:* (a) Photograph detail blurred with discontinuous kernel from Figure 6.1(d). – (b) Linear deblurring by Wiener filter. – (c) Diffusion–reaction deblurring with TV diffusivity. – (d) Diffusion–reaction deblurring with Perona–Malik diffusivity. In (b)–(d), a continuous periodic extrapolation of the blurred image was used (realistic method). While details are well reconstructed, shadow-like boundary artifacts affect the overall quality. *Bottom row (e)–(h):* Same as (a)–(d) but with special setting to suppress boundary artifacts. The image was extended by reflection to four times its original size *before* blurring. This quadruple-size blurred image was then blurred and deblurred with periodic continuation. – Images: D. Theis, from [222].

In Figure 6.5, a more detailed comparison of deblurring algorithms is presented for a detail from the photograph blurred with the discontinuous kernel, including Wiener filter as an example of linear deblurring, diffusion–reaction filtering with TV, and Perona–Malik diffusivity. Here we also demonstrate the use of our special test setting to avoid boundary artifacts in quality measurements.

Blurred *MIA Letters* image. Results for the grey-value test image are shown in Figure 6.6. Here, we concentrate on the discontinuous kernel from Figure 6.1(d). Besides the methods mentioned above we show also diffusion–reaction deblurring with anisotropic diffusion tensor which performs particularly well for this type of strongly segmented images.

Figure 6.7 shows a detail of our grey-value test image to demonstrate the improvements made by anisotropic diffusion tensors and continuation strategy. Perona–Malik deblurring with a single fixed $\alpha > 0$ reduces oscillatory artifacts quite well but smears thin lines while the fixed-point iteration with $\alpha = 0$ restores many details but generates artifacts similar to those of linear deconvolution. The continuation



Figure 6.6: *Top left (a)*: Grey-value *MIA* Letters test image. *Top middle (b)*: Blurred with discontinuous kernel from Figure 6.1(d). *Top right (c)*: Linear deblurring by Wiener filter, boundary treatment by continuous extrapolation. *Bottom left (d)*: Diffusion-reaction deblurring with TV diffusivity and continuation strategy (2 levels). *Bottom middle (e)*: Same with Perona–Malik diffusivity. *Bottom right (f)*: Same with anisotropic diffusion tensor. – From [222]. Deconvolved images: D. Theis.

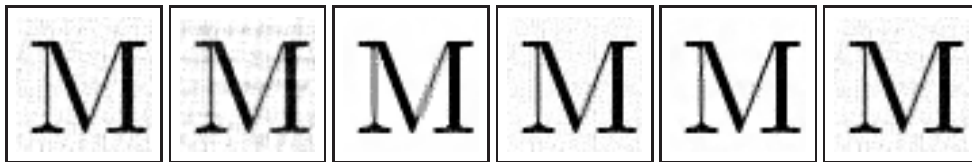


Figure 6.7: Detail from deblurred *MIA* Letters images, with boundary artifacts suppressed by special test setting. *Left to right: (a)* Linear filtering. – *(b)* Unregularised variational model. – *(c)* Perona–Malik, constant diffusion weight. – *(d)* Perona–Malik followed by nonregularised iteration (two-step continuation strategy). – *(e)* Perona–Malik, continuation strategy with 10 steps. – *(f)* Anisotropic diffusion-reaction, continuation strategy with 10 steps. – From [222].

Table 6.1: SNR (dB) for deblurring with the discontinuous kernel. First values: specific test setting for boundary conditions, second values: with continuous extrapolation. From [222].

	Wiener filtering	Diffusion-reaction, Perona–Malik	Diffusion-reaction, TV diffusivity
Letters	15.6 / 7.1	18.4 / 7.3	17.1 / 7.2
– with contin. strategy		19.3 / 7.4	19.2 / 7.1
Photograph detail	15.9 / 9.4	14.9 / 6.0	14.3 / 5.9

strategy combines a better restoration of details with a reasonable suppression of artifacts. Even in its simplest form with two steps it bears a clear improvement; more steps lead to further enhancement. The sharpness of edges is further improved by using the anisotropic diffusion tensor.

In Table 6.1 we compile measurements of the signal-to-noise ratio (SNR)

$$\text{SNR}(v, u) = 10 \log_{10} \frac{\text{var}(u)}{\text{var}(u - v)} \text{ dB} \quad (6.55)$$

where u is the original image and v the deblurring result. Throughout the measurements Perona–Malik deblurring tends to slightly better SNR than deblurring with TV diffusivity. However, not always do SNR measurements reflect sufficiently well visual judgement. For the photograph, e.g., the Wiener filter performs better than diffusion–reaction deblurring in terms of SNR. On the other hand, Figure 6.5 clearly reveals the superiority of diffusion–reaction deblurring.

6.6.2 Deblurring of Photographs Blurred During Exposure

Our second group of experiments aims at deblurring images which have been blurred during exposure by camera movement or defocussing. In these tests, we used the gradient descent PDE resulting from a variational approach with robust L^1 data term and Perona–Malik regulariser. In Figure 6.8, we also show results with non-robust (quadratic) data term and quadratic regulariser for comparison.

A caveat about all comparisons between methods presented here is that the regularisation weight α cannot be compared between methods because different penalisers differ in the magnitudes of their energy or gradient descent contributions.

Photograph of Paris (camera motion). In Figure 6.8(a), a photograph of Paris at dusk is shown which has been blurred by camera movement during exposure. The PSF is fairly irregular. For the deblurring process, it was assumed to be spatially invariant. By clipping the image of an isolated light source from the lower part of the river region, the approximate PSF shown in Figure 6.1(e) was obtained and used for deblurring.

Clearly, the estimation of the convolution kernel from such an approximate impulse response induces an imprecision. Further, a closer look at the blurred image reveals that the assumption of spatially invariant blur does not perfectly capture the situation since impulse responses in opposite corners of the image are of slightly different shape.

The deconvolution result obtained with quadratic data terms and regularisers, Figure 6.8(b), shows that while the image can be sharpened to some degree but fine details, like the small light sources, are visibly smeared out due to the linear diffusion contribution. At the same time, strong oscillatory artifacts can be seen that prevent reducing the regularisation weight (and thereby the smearing).



Figure 6.8: *Top left (a)*: Paris at dusk from Eiffel tower, blurred by camera movement during exposition (480×480 pixels). – *Top right (b)*: Restoration by variational deblurring with quadratic data term and quadratic Tikhonov regulariser, $\alpha = 0.05$. The spatially invariant, irregularly shaped PSF from Figure 6.1(e) was used as approximate PSF. – *Bottom left (c)*: Same with quadratic data term and edge-enhancing Perona–Malik regulariser, $\lambda = 26$, $\alpha = 1.5$. – *Bottom right (d)*: Variationally deblurred with robust (L^1) data term and Perona–Malik regulariser, $\lambda = 26$, $\alpha = 0.06$. – Be aware that values of the regularisation weight α cannot be compared between different methods. – Partially following [223].

Combining an edge-enhancing Perona–Malik regulariser with non-robust data terms increases the sharpness of fine details, Figure 6.8(c). To keep oscillatory artifacts low, it was still necessary to choose a large diffusion weight which caused most finer structures in the image to be smoothed away. Nevertheless, street lights are still restored inaccurately, as they are accompanied by shadows and echo images. Street lights re-



Figure 6.9: Enlarged detail views (240×117 pixels) from the images shown in Figure 6.8, middle right region. *Top left (a)*: Original image. – *Top right (b)*: Quadratic data term, quadratic regulariser. – *Bottom left (c)*: Quadratic data term, Perona–Malik regulariser. – *Bottom right (d)*: Robust data term, Perona–Malik regulariser. – The bridge region chosen is close to the location from which the PSF has been extracted. Increasing sharpness and reduction of artifacts can be seen. – Partially following [223].

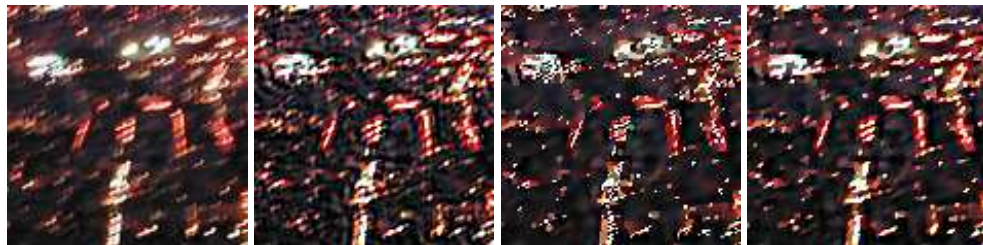


Figure 6.10: Enlarged detail views (117×117 pixels) from the images shown in Figure 6.8, top left region. *Left to right: (a)* Original image. – *(b)* Quadratic data term, quadratic regulariser. – *(c)* Quadratic data term, Perona–Malik regulariser. – *(d)* Robust data term, Perona–Malik regulariser. – As the region shown here is remote from the location where the PSF has been extracted, the PSF estimate is rather inaccurate here. Deconvolution with non-robust data terms performs poor while robust deconvolution achieves a considerable quality. – Partially following [223].

note from the river region are particularly poorly restored because of the PSF inaccuracy.

Robust deblurring, in contrast, can cope with the PSF imprecision much better, see Figure 6.8(d). The strength of regularisation could be reduced such that more structures like streets and buildings are recovered. Shadows and double images near the



Figure 6.11: Enlarged detail views (117×117 pixels) from the images shown in Figure 6.8, bottom left region, with linearly enhanced contrast. *Left to right: (a)* Original image. – *(b)* Quadratic data term, quadratic regulariser. – *(c)* Quadratic data term, Perona–Malik regulariser. – *(d)* Robust data term, Perona–Malik regulariser. – In this low-contrast region, the deconvolution result (b) is dominated by artifacts while in (c) the strong regularisation needed leads to a partial loss of details like the building contours. The reduced amount of regularisation with robust data terms (d) allows for a better preservation of these features. – Partially following [223].

street lights do hardly occur. Even in the lower left part of the image where the PSF shape deviates much from that in the river region favourable sharpness is achieved.

Figures 6.9–6.11 show clippings from the same images in which these effects can be seen clearer.

Printed text (spatially variant defocus). The last experiment of this chapter, Figure 6.12, emphasises the spatial variation of the PSF. A piece of printed text was photographed from small distance without appropriate focussing. The distance between lens and object varied widely, leading to a stronger defocussing in the lower than in the upper part of the image. Theoretical considerations show that defocussing PSFs are well approximated by cylindrical functions (normalised characteristic functions of disks, compare Figure 6.1(b)). For defocussing we used therefore a cylindrical PSF whose diameter varies linearly from 5 at the top edge to 10.5 at the bottom edge.

To demonstrate also the robustness of the proposed deblurring method with respect to noise, we replaced 30% of all pixels by uniform noise. Noise of such intensity is not typically encountered in application data that the method is intended for, so we had to resort to artificial image degradation at this single point. Restoration using our variational method still works well. One difference is that the continuation strategy must stop reducing the diffusion weight α at a larger value now, in order to remove the oversmoothing at the initial large α while preserving the noise suppression by regularisation.

We stress that although we used in Figure 6.12 a special PSF shape where effectively only one parameter – the diameter – controls the spatial variation, our deblurring

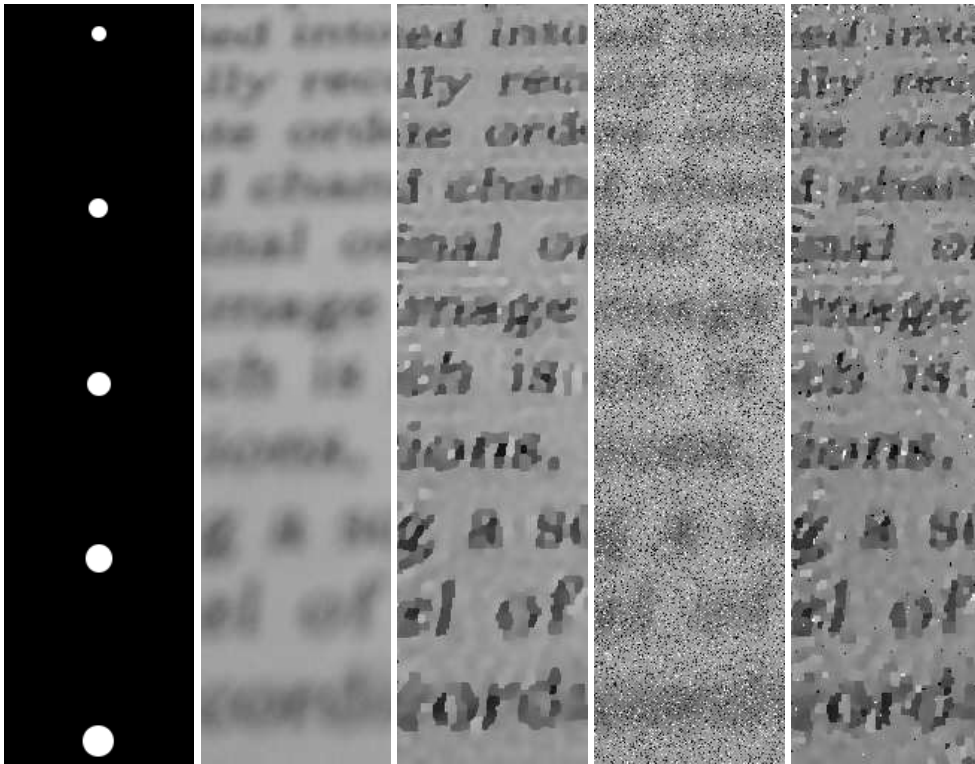


Figure 6.12: *Left to right: (a) Spatially variant defocussing PSF (correct size). – (b) Defocussed photograph of printed text. – (c) Restored by variational deblurring with robust data term and Perona–Malik regulariser ($\lambda = 5$), using continuation strategy (2 levels with $\alpha = 0.03$ and $\alpha = 0.003$). – (d) Defocussed photograph with 30% uniform noise. – (e) Restored by variational deblurring with robust data term and Perona–Malik regulariser ($\lambda = 5$), using continuation strategy (2 levels with $\alpha = 0.03$ and $\alpha = 0.015$). – From [223].*

PDE itself is not restricted to such a setting. Indeed it can cope with arbitrarily given $H(x, y)$.

Chapter 7

Inequality Constraints in Variational Deconvolution

In this chapter we will extend the basic variational framework for deblurring developed in the previous chapter by introducing inequality constraints.

Since typical deconvolution problems are ill-posed inverse problems, it is highly desirable to use any additional information that is available to support the sharpening process. Inequality constraints are one type of such additional information which can often be derived e.g. from physical considerations: In a grey-value image whose values are proportional to radiance, these values are bounded from below since radiance cannot take negative values. Parameters of the image acquisition process sometimes allow also to derive an upper bound. A similar situation occurs in the context of diffusion tensor magnetic resonance imaging (DTMRI). The physical nature of the measured diffusion tensors implies that they must be positive (semi-)definite, which is an inequality constraint, too.

The principal idea of our approach to variational deconvolution with inequality constraints consists in a reparametrisation of the image range. In the grey-value case, the reparametrisation principle has been used before in the context of a discrete deconvolution model [151]. Moreover, the reparametrisation admits an interesting differential geometric reinterpretation which subsequently enables a transfer of the technique to more complex inequality constraints on multi-channel images, such as positive definiteness of matrix-valued data. We will demonstrate the capabilities of the approach by experiments on a colour photograph and on DTMRI data.

The presentation of the concepts and results in this chapter follows the conference paper published at IbPRIA 2007 [220].

Related work. Reparametrisation has been used to impose a positivity constraint on a discrete deconvolution model in [151]. A differential geometric framework for gradient descent constrained to submanifolds (i.e., equality constraints) has been discussed in [64].

7.1 Basic Framework

We start from a model for variational deconvolution of (single- or) multi-channel images as described in Chapter 6 and use all notations from there. The energy functional for a model with robust data terms (joint robustification, compare the discussion on page 226) and edge-enhancing regulariser reads

$$E[u] = \int_{\Omega} \left(\Phi \left(\sum_{k \in \Gamma} (R_{f_k, H}[u_k])^2 \right) + \alpha \Psi \left(\sum_{k \in \Gamma} |\nabla u_k|^2 \right) \right) dx \quad (7.1)$$

where the index set Γ enumerates the image channels ($|\Gamma| = 1$ for grey-value images).

A gradient descent for (7.1) is given by

$$\begin{aligned} \partial_t u_k = G_1 := & \left(\int_{\Omega} \Phi' \left(\sum_{l \in \Gamma} (R_{f_l, H}[u_l](y))^2 \right) (R_{f_k, H}[u_k](y)) H(y, \cdot) dy \right) \\ & + \alpha \operatorname{div} \left(\Psi' \left(\sum_{l \in \Gamma} |\nabla u_l|^2 \right) \nabla u_k \right). \end{aligned} \quad (7.2)$$

We next consider how the model (7.1) can be modified to incorporate constraints on the solution.

7.1.1 Constraints for Greyvalue Images

Let us assume first that we are dealing with the single channel case, where the pixels represent grey-value intensities. Since negative intensities do not physically make sense, we would like to modify (7.1) to constrain the grey-values to be nonnegative.

One possibility would be to add a penalty for negative values, Such a “soft constraint” would harmonise with the structure of the variational approach. However, it brings about an additional weight parameter which is difficult to tune, and it also does not strictly enforce the inequality.

7.1.1.1 Reparametrisation Approach

Another possibility, which has been shown to be very effective for discrete deconvolution problems [151], reparametrises the greyvalues via $u = \exp(z)$ with a new

image function z whose values are unconstrained in \mathbb{R} , and calculates the gradient descent for z . Slightly generalising, we substitute $u = \varphi(z)$ with a smooth invertible function $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ into (7.1) and obtain

$$\tilde{E}[z] = \frac{1}{2} \int_{\Omega} \left(\Phi \left((R_{f,H}[\varphi \circ z](x))^2 \right) + \alpha \Psi \left((\varphi'(z(x)) |\nabla z(x)|)^2 \right) \right) dx. \quad (7.3)$$

7.1.1.2 Gradient Descent Calculation

We derive now the variational derivative of the energy functional with respect to z . For convenience, we decompose the energy functional again into data term and regulariser.

Variation of data term. Inserting into the data term

$$\tilde{E}_1[z] := \frac{1}{2} \int_{\Omega} \Phi \left((R_{f,H}[\varphi \circ z](x))^2 \right) dx \quad (7.4)$$

for z the function $w = z + \varepsilon v$, where v is assumed to be a smooth function on Ω which vanishes on the boundary, we calculate

$$\begin{aligned} \left. \frac{d}{d\varepsilon} \tilde{E}_1[w] \right|_{\varepsilon=0} &= \left. \frac{d}{d\varepsilon} \left(\frac{1}{2} \int_{\Omega} \Phi \left((R_{f,H}[\varphi \circ w](x))^2 \right) dx \right) \right|_{\varepsilon=0} \\ &= \int_{\Omega} \left(\Phi' \left((R_{f,H}[\varphi \circ z](y))^2 \right) R_{f,H}[\varphi \circ z](y) \right) \cdot \\ &\quad \cdot \left. \frac{d}{d\varepsilon} \left(f(y) - \int_{\Omega} \varphi(z(x) + \varepsilon v(x)) H(y, x) dx \right) \right|_{\varepsilon=0} dy \\ &= - \int_{\Omega} \left(\Phi' \left((R_{f,H}[\varphi \circ z](y))^2 \right) R_{f,H}[\varphi \circ z](y) \right) \cdot \\ &\quad \cdot \int_{\Omega} \varphi'(z(x)) v(x) H(y, x) dx dy \\ &= - \int_{\Omega} \int_{\Omega} \left(\Phi' \left((R_{f,H}[\varphi \circ z](y))^2 \right) R_{f,H}[\varphi \circ z](y) \right) \cdot H(y, x) dy \cdot \\ &\quad \cdot \varphi'(z(x)) \cdot v(x) dx. \end{aligned} \quad (7.5)$$

As this equality has to hold for all admissible functions v , we drop the integral and multiplication with v and obtain as variational derivative of \tilde{E}_1 the expression

$$\begin{aligned} \frac{\delta}{\delta z} \tilde{E}_1[z](x) &= - \int_{\Omega} \left(\Phi'((R_{f,H}[\varphi \circ z](y))^2) R_{f,H}[\varphi \circ z](y) \right) \cdot H(y, x) dy \cdot \varphi'(z(x)) \quad (7.6) \end{aligned}$$

Substituting $z := \varphi^{-1}(u)$ with the inverse function φ^{-1} , the variational gradient is expressed in terms of the true image data u as

$$\begin{aligned} \frac{\delta}{\delta z} \tilde{E}_1[z](x) &= - \int_{\Omega} \left(\Phi'((R_{f,H}[u](y))^2) R_{f,H}[u](y) \right) \cdot H(y, x) dy \cdot \varphi'(\varphi^{-1}(u(x))). \quad (7.7) \end{aligned}$$

Variation of the regulariser. Turning to the regulariser

$$\tilde{E}_2[z] = \frac{1}{2} \int_{\Omega} \Psi\left(\left(\varphi'(z(x))|\nabla z(x)|\right)^2\right) dx \quad (7.8)$$

and substituting for z again $w = z + \varepsilon v$ with a smooth function v on Ω which vanishes on the boundary, we have

$$\begin{aligned} \left. \frac{d}{d\varepsilon} \tilde{E}_2[w] \right|_{\varepsilon=0} &= \left. \frac{d}{d\varepsilon} \left(\frac{1}{2} \int_{\Omega} \Psi\left(\left(\varphi'(w)|\nabla w|\right)^2\right) dx \right) \right|_{\varepsilon=0} \\ &= \frac{1}{2} \int_{\Omega} \Psi'\left(\left(\varphi'(z)|\nabla z|\right)^2\right) \cdot \sum_i \left. \frac{d}{d\varepsilon} \left(\varphi'(z + \varepsilon v) \partial_{x_i}(z + \varepsilon v) \right)^2 \right|_{\varepsilon=0} dx \\ &= \frac{1}{2} \int_{\Omega} \Psi'\left(\left(\varphi'(z)|\nabla z|\right)^2\right) \cdot \\ &\quad \cdot \sum_i \left(2\varphi'(z) \cdot \varphi''(z) \cdot \left. \frac{d}{d\varepsilon} (z + \varepsilon v) \right|_{\varepsilon=0} \cdot (\partial_{x_i} z)^2 \right. \\ &\quad \left. + (\varphi'(z))^2 \cdot \left. \frac{d}{d\varepsilon} \left((\partial_{x_i} z)^2 + 2\varepsilon \partial_{x_i} z \cdot \partial_{x_i} v + \varepsilon^2 (\partial_{x_i} v)^2 \right) \right|_{\varepsilon=0} \right) dx \\ &= \int_{\Omega} \Psi'\left(\left(\varphi'(z)|\nabla z|\right)^2\right) \cdot \sum_i \left(\varphi' \cdot \varphi'' \cdot (\partial_{x_i} z)^2 \cdot v + \varphi'^2 \cdot \partial_{x_i} z \cdot \partial_{x_i} v \right) dx. \quad (7.9) \end{aligned}$$

Integration by parts yields

$$\begin{aligned}
& \int_{\Omega} \Psi' \left((\varphi'(z) |\nabla z|)^2 \right) \cdot (\varphi'(z))^2 \cdot \partial_{x_i} z \cdot \partial_{x_i} v \, dx \\
&= - \int_{\Omega} \partial_{x_i} \left(\varphi'(z) \right) \cdot \varphi'(z) \cdot \Psi' \left((\varphi'(z) |\nabla z|)^2 \right) \cdot \partial_{x_i} z \cdot v \, dx \\
&\quad - \int_{\Omega} \varphi'(z) \cdot \partial_{x_i} \left(\varphi'(z) \cdot \Psi' \left((\varphi'(z) |\nabla z|)^2 \right) \cdot \partial_{x_i} z \right) \cdot v \, dx \quad (7.10)
\end{aligned}$$

which by inserting into (7.9) leads to

$$\left. \frac{d}{d\varepsilon} \tilde{E}_2[w] \right|_{\varepsilon=0} = - \int_{\Omega} \operatorname{div} \left(\Psi' \left((\varphi'(z) |\nabla z|)^2 \right) \cdot \varphi'(z) \nabla z \right) \cdot \varphi'(z) \cdot v \, dx \quad (7.11)$$

and by back-substitution $z = \varphi^{-1}(u)$ to

$$\left. \frac{d}{d\varepsilon} \tilde{E}_2[w] \right|_{\varepsilon=0} = - \int_{\Omega} \operatorname{div} \left(\Psi' (|\nabla u|^2) \cdot \nabla u \right) \cdot \varphi'(\varphi^{-1}(u)) \cdot v \, dx \quad (7.12)$$

Dropping as before the integral and multiplication by v , we arrive at the variational gradient

$$\frac{\delta}{\delta z} \tilde{E}_2[z] = - \operatorname{div} \left(\Psi' (|\nabla u|^2) \nabla u \right) \cdot \varphi'(\varphi^{-1}(u)). \quad (7.13)$$

Combining (7.7) and (7.13), we find the new gradient descent

$$\begin{aligned}
\partial_t u = G_{\varphi} := & \varphi'(\varphi^{-1}(u)) \cdot \left(\int_{\Omega} \Phi' \left((R_{f,H}[u](y))^2 \right) (R_{f,H}[u](y)) H(y, \cdot) \, dy \right) \\
& + \alpha \operatorname{div} \left(\Psi' (|\nabla u|^2) \nabla u_k \right) \quad (7.14)
\end{aligned}$$

which differs from (7.2) (with $|\Gamma| = 1$) only by a factor on the right-hand side, i.e., we have

$$G_{\varphi} = \varphi'(\varphi^{-1}(u)) \cdot G_1. \quad (7.15)$$

We will refer to the factor $\varphi'(\varphi^{-1}(u))$ as *constraint multiplier*.

7.1.1.3 Specific Reparametrisations

It remains to specify suitable functions φ which can realise via reparametrisation the desired inequality constraints, and to calculate the corresponding constraint multipliers.

Positivity constraint. A positivity constraint is imposed by

$$\varphi(z) = \exp(z), \quad (7.16)$$

with the constraint multiplier

$$\varphi'(\varphi^{-1}(u)) = u. \quad (7.17)$$

This reparametrisation has been introduced in [151].

Interval constraint. The positivity constraint can easily be generalised to an interval constraint $a < u < b$ by using a sigmoid function such as

$$\varphi(z) = \frac{a \exp(-z) + b}{\exp(-z) + 1}, \quad (7.18)$$

leading to the constraint multiplier

$$\varphi'(\varphi^{-1}(u)) = \frac{(u-a)(b-u)}{b-a}. \quad (7.19)$$

7.1.2 Separable Constraints for Multi-Channel Images

Going from a single grey-value to multi-channel images, the simplest type of constraints that comes into sight are *separable* inequality constraints, i.e., constraints that are represented by independent inequalities on the individual channels. The extension of the framework developed above for single-channel images to this situation is obvious.

Constraining Colour Images. For example, positivity or interval constraints for the channels of colour images are imposed by setting $u_k = \varphi_k(z_k)$ for $k \in \Gamma$. The corresponding gradient descent in channel k is given by equation (7.2) with the right-hand side multiplied by $\varphi_k^{-1}(\varphi_k'(u_k))$.

Experiment. In our first experiment, we consider deconvolution of a colour image with positivity constraint in all channels using the gradient descent (7.14) with $\varphi(z) = \exp(z)$. We use robust L^1 data terms in combination with an edge-enhancing Perona-Malik regulariser.

As in the case of the unconstrained gradient descent (7.2), a straight forward numerical implementation is through an explicit time-stepping scheme which is stopped when the updates fall below some small positive threshold. We do not focus on experiments with additional noise since the robustness of variational deconvolution



Figure 7.1: Variational deconvolution of the colour photograph from Figure 6.8(a), using the approximate space-invariant PSF from Figure 6.1(e). In all cases, robust L^1 data terms and Perona–Malik regularisers with $\lambda = 26$ have been used. In this setting, regularisation weights can be compared. *Top left (a)*: Without constraint, $\alpha = 0.06$ (same as Figure 6.8(d)). – *Top right (b)*: With positivity constraint, $\alpha = 0.06$. – *Bottom left (c)*: Without constraint, $\alpha = 0.03$. – *Bottom right (d)*: With positivity constraint, $\alpha = 0.03$. – Following [220].



Figure 7.2: Details (240×117 pixels) from the images shown in Figure 7.1, middle right region. *Top left (a)*: No constraint, $\alpha = 0.06$ – *Top right (b)*: Positivity constraint, $\alpha = 0.06$. – *Bottom left (c)*: No constraint, $\alpha = 0.03$. While light spots are still more blurry than in (b), artifacts are amplified, particularly near the riverbank. – *Bottom right (d)*: Positivity constraint, $\alpha = 0.03$. Artifacts are suppressed similarly as in (a) but light spots as well as the light bands at the bridge appear sharper than in (b) and (c). – Following [220].



Figure 7.3: Details (117×117 pixels each) from the images shown in Figure 7.1. *Top row (a)–(d)*: Detail from top left region as in Figure 6.10. *Bottom row (e)–(h)*: Detail from bottom left region, with linearly enhanced contrast as in Figure 6.11. *From left to right*: (a)/(e) No constraint, $\alpha = 0.06$. – (b)/(f) Positivity constraint, $\alpha = 0.06$. A slight gain in the sharpness of some details is visible. – (c)/(g) No constraint, $\alpha = 0.03$. Many details that were smoothed away in (a)/(e) become distinguishable but oscillations on the finest scale appear close to light sources (black pixels). – (d)/(h) Positivity constraint, $\alpha = 0.03$. Sharpness is comparable to (c)/(g) but the artifacts are suppressed.

with robust data terms under e.g. impulsive noise has already been demonstrated in Section 6.6.2, Figure 6.12.

Figure 7.1 shows deconvolution results without constraints and with positivity constraints for the blurred colour photograph of Paris from Figure 6.8(a). Since in both cases the same energy functional is minimised, the comparison of regularisation weights α is meaningful here. Remember that the exact PSF is unknown and slightly space-variant while a space-invariant PSF estimated from an impulse response in the image is used for deconvolution.

In Figure 7.1(a) we show again the result from Figure 6.8(d), i.e., deconvolution without constraint, $\alpha = 0.06$. In (b) a positivity constraint is used while retaining $\alpha = 0.06$. Already here, a gain in sharpness of small details can be observed. Reducing the regularisation weight in both settings to $\alpha = 0.03$ leads to the results shown in (c) and (d). Without constraint (c), a slight gain in sharpness is accompanied by a noticeable increase of oscillatory artifacts, best visible near edges. In contrast, the result with positivity constraint (d) still shows a good suppression of artifacts. Interestingly, this includes not only undershoots to negative values which are suppressed directly, but also overshoots in the positive range, due to the convolution in the data term that links over- and undershoots. The overall level of artifacts in (d) is comparable to that in (a), while fine image structures are reconstructed much better. In practice, the positivity constraint thus allows to reduce the degree of regularisation and thereby to improve the deconvolution of details.

This comparison is further illustrated by Figures 7.2 and 7.3 which show clippings from the images of Figure 6.8.

7.2 Geometric Reinterpretation

We now show that it is possible to interpret the reparametrisation approach geometrically, and this leads to a very convenient form in which more general constraints can be easily incorporated into the model.

Positivity constraint and hyperbolic metric. Let us consider once more grey-value images with positivity constraint, i.e., $\varphi'(\varphi^{-1}(u)) = u$. We have obtained (7.14) as the gradient descent for the functional $\tilde{E}[z]$. This derivation implicitly refers to an inner product in a function space (compare our remark in 2.1.2 after (2.8)), namely that of the variation functions $v = \delta z$. The inner product we have used is the standard inner product

$$\langle \delta z_1, \delta z_2 \rangle = \int_{\Omega} \delta z_1(x) \delta z_2(x) dx \quad (7.20)$$

which is constructed from the standard (Euclidean) metric dz on the range of $z : \Omega \rightarrow \mathbb{R}$.

Let us now consider the range of positive grey-values \mathbb{R}^+ , i.e., the range of u , and replace the Euclidean metric du with the *hyperbolic metric* given by

$$d_h u := \frac{du}{u}. \quad (7.21)$$

This implies also a different metric on the space of functions $u : \Omega \rightarrow \mathbb{R}^+$, which is expressed by

$$\delta_h u := \frac{\delta u}{u}, \quad (7.22)$$

and thereby to a different inner product on its tangential space, the space of variation functions δu . This inner product is given by

$$\langle \delta u_1, \delta u_2 \rangle_h := \langle \delta_h u_1, \delta_h u_2 \rangle \quad (7.23)$$

We will now see that the gradient descent (7.14) also arises as gradient descent for the *original* energy functional $E[u]$ from (7.1) w.r.t. the inner product (7.23). Variations of data term and regulariser have been computed before as (6.14) and (2.24), resp., and sum up to

$$\delta E = - \int_{\Omega} G_1(x) \cdot v(x) dx \quad (7.24)$$

where we have used the abbreviation G_1 as defined in (7.2). According to 2.1.2 the sought variational gradient is the function G for which

$$\delta E = \langle -G, \delta_h u \rangle \quad (7.25)$$

which, by comparison to $\delta E = \langle -G_1, \delta u \rangle$ implies $G = u \cdot G_1$ in accordance with (7.19) for $\varphi \equiv \exp$.

We can therefore represent our modified gradient descent process as standard gradient descent with a different underlying metric! From this viewpoint, zero and negative greyvalues are avoided simply because the hyperbolic metric puts them at infinite distance from any positive values.

This reinterpretation can directly be transferred to the interval constraint case, by choosing a metric on (a, b) in which both interval ends a and b are pushed away to infinite distance.

7.3 Positive Definiteness as an Inequality Constraint

As a consequence of our geometric reinterpretation, we no longer need to rely on an explicit reparametrisation of the image range to compute a constrained gradient

descent. Instead, it is sufficient to calculate the gradient descent of (7.1) with respect to a suitably chosen metric on the image range. We carry this out for one example.

Positive definiteness constraint in matrix-valued images. We consider matrix-valued images as in Chapter 3. As pointed out there positive definiteness is an essential property of such data in several relevant application contexts. We want therefore to formulate a gradient descent for variational deconvolution of matrix-valued images with positive definiteness constraint.

To this end, we use the Riemannian metric on the cone of positive definite matrices that is given by $d_S A^2 = \|A^{-1/2} dA A^{-1/2}\|_F^2$ with $\|\cdot\|_F$ denoting the Frobenius norm, see [187, 106]. This metric has recently been investigated intensively in the context of DTMRI data processing, see e.g. [143, 160].

Denoting the matrix-valued image by $U = (u_k)_{k \in \Gamma}$, $\Gamma = \{1, 2, 3\} \times \{1, 2, 3\}$, the gradient descent for (7.1) with respect to the metric d_S is given by

$$\partial_t U = U^{1/2} G U^{1/2} \quad (7.26)$$

where G is the matrix of all the right-hand sides of (7.2) for channels $k \in \Gamma$.

Experiment with DT-MRI data. In Figure 7.4 we demonstrate deconvolution of matrix-valued data with positive definiteness constraint, using data from diffusion tensor MRI measurements. We employ once more robust L^1 data terms, an edge-enhancing Perona–Malik regulariser and compute the gradient descent according to (7.26).

Figure 7.4 (a), (b) show one 2D slice of the DTMRI data set that has already been used in Chapter 3, see e.g. Figure 3.4. In (c), (d) it has been synthetically blurred by iterative box filtering approximating a Gaussian of standard deviation 2, and in (e), (f) deconvolved by our method, using a Gaussian PSF of the same standard deviation.

It can be seen that many structures in the DTMRI image are nicely reconstructed by the deconvolution process, e.g. the highly anisotropic diffusion tensors of the corpus callosum region, and the sharp edges between corpus callosum and the large isotropic tensors in the adjacent ventricle. A limitation of the current method that can be seen from the figures is that very thin details with a width of only one or two voxels are still smoothed in the deconvolved image. Due to the low resolution this effect is more relevant in DTMRI data than elsewhere, and further work will be devoted to improve reconstruction quality for such details.

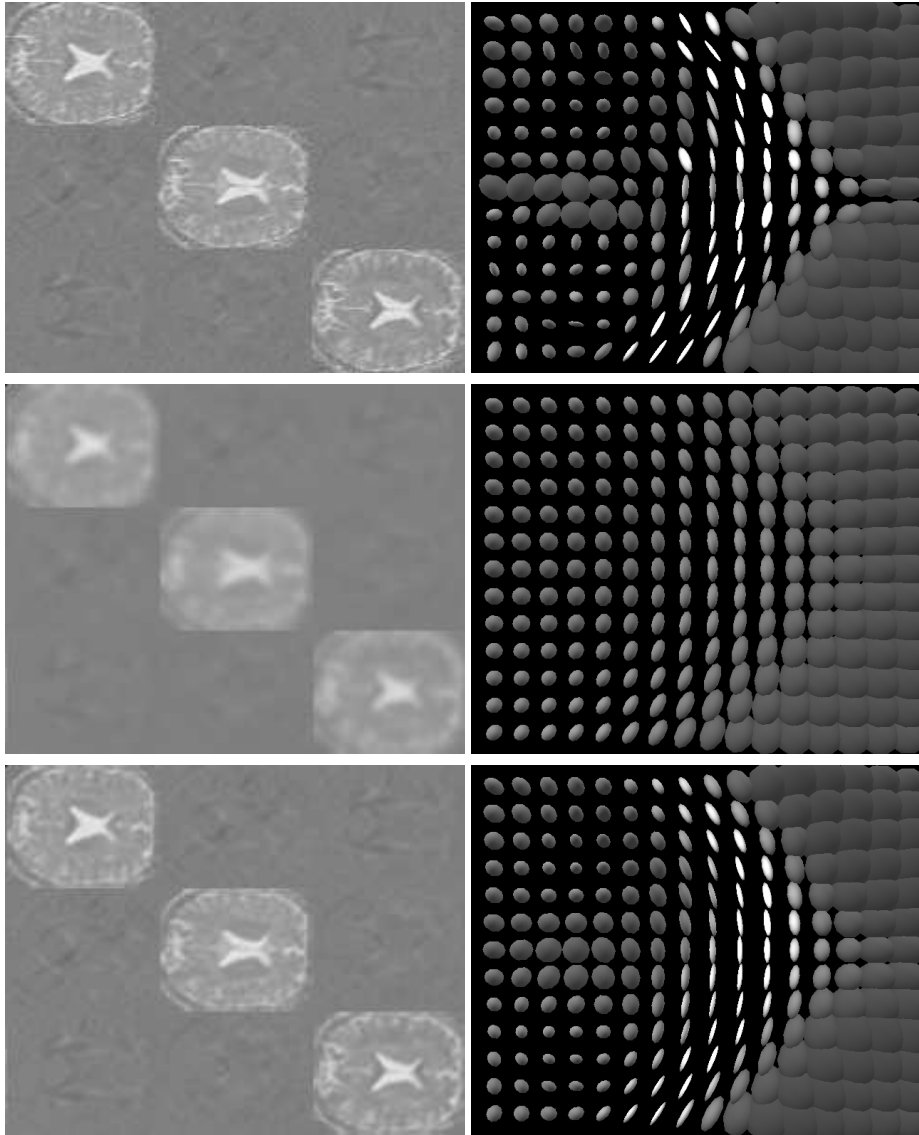


Figure 7.4: *Top left (a)*: One 2D slice from a DT-MRI data set of a human brain. The 3×3 tiles represent the matrix components, with middle grey representing 0. *Top right (b)*: Detail from the corpus callosum region visualised by ellipsoids. Directions and lengths of the principal axes correspond to eigenvectors and eigenvalues, resp. *Middle row (c), (d)*: Blurred by iterated box filtering, approximating convolution with a Gaussian of standard deviation 2. *Bottom row (e), (f)*: Variational deconvolution with robust data terms, total variation regulariser, regularisation weight 0.03, and positive definiteness constraint. – From [220].

Concluding Remarks

We have presented theoretical developments and experimental results on different mathematical models and algorithms for the processing of digital single- and multi-channel images. One common theme behind the contributions included in this work is given by the use of geometric concepts and invariances as guiding principle in the design of models and algorithms. Another common theme is that image filtering processes that involve an evolution from the original to the filtered image can be seen as dynamical systems. Both principles were made useful in a number of ways.

We have detailed the development of a stringent theory for median filters on multi-channel images with focus on matrix-valued images, based on a geometrically motivated axiomatic approach. Extensions included several median-related filters and particularly a family of M-smoothers with power laws for the penalisation of differences.

Further we have studied space-discrete versions of PDE-based signal and image filters using dynamical systems of ODEs. The selection of filters covered on one hand the family of diffusion filters including linear, nonlinear isotropic and nonlinear anisotropic diffusion, in particular also singular isotropic processes like total variation flow. On the other hand exemplary structure-enhancing filters – a shock filter of Osher-Rudin type, stabilised inverse linear diffusion, and the so-called forward-and-backward diffusion were considered and important aspects analysed. Due to the occurrence of singularities in several of these processes, techniques for dynamical systems with discontinuous right-hand sides had to be used. For all types of diffusion filters, we also were able to devise numerical schemes of special kind, namely, locally (semi-) analytic schemes, with interesting stability properties, low numerical dissipation, and simple algorithmic structure. These algorithms also constituted a bridge between diffusion and wavelet-based image denoising filters.

In the last part we have discussed a variational model framework for the sharpening of images degraded by blur. The models presented here are characterised by excellent robustness against noise and violations of model assumptions. They admit fairly general blur models: space-variant point-spread functions, even of fairly irregular shape. By their continuous-domain design, they also obey the geometric invariance principles

that are essential in image processing. Finally, we showed how inequality constraints can be integrated into the variational framework. The way to do this is again founded on a geometric construction, this time concerning the metrics in the function space underlying the gradient descent for the energy functional.

Representing current research, this work naturally remains incomplete in many respects, leaving open questions. Let us outline a few major directions of ongoing research.

The investigation of spatially discretised PDE filters calls for extension to further classes of filters. In the case of the structure enhancement filters from Chapter 5, also the extension to the 2D case is awaiting research. Further it can be hoped that also the idea of locally (semi-) analytic schemes can be transferred, e.g., to curvature-based PDE filters like mean curvature motion (compare 3.1.1, 3.5.1).

In the case of variational deconvolution, algorithmic improvements are one desideratum. This includes more efficient numerics for the gradient descent equations, and possibly also other ways to minimise the energy functionals in question. Also, the stability analysis for the explicit numerical schemes started in Chapter 6 will be continued and extended to further schemes.

Last but not least, experience shows that deconvolution methods that work fairly well on synthetic examples can still fail on naturally blurred images. We have therefore consciously carried out many experiments here on physically blurred images. However, proper benchmarking becomes difficult – currently, mostly impossible – in this way. A further analysis of the physical blur process and creation of better benchmarks will be necessary to replace questionable visual judgements.

Bibliography

- [1] R. Acar, C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10:1217–1229, 1994
- [2] D. Alexander. An introduction to computational diffusion MRI: the diffusion tensor and beyond. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 83–106, Berlin, 2006, Springer
- [3] L. Alvarez, F. Guichard, P.-L. Lions, J.-M. Morel. Axioms and fundamental equations in image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, 1993
- [4] L. Alvarez, P.-L. Lions, J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. II. *SIAM Journal on Numerical Analysis*, 29:845–866, 1992
- [5] L. Alvarez, L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, 31:590–605, 1994
- [6] F. Andreu, C. Ballester, V. Caselles, J. M. Mazón. Minimizing total variation flow. *Differential and Integral Equations*, 14(3):321–360, 2001
- [7] F. Andreu, V. Caselles, J. I. Diaz, J. M. Mazón. Some qualitative properties of the total variation flow. *Journal of Functional Analysis*, 188(2):516–547, 2002
- [8] F. Andreu-Vaillio, V. Caselles, J. M. Mazón. *Parabolic Quasilinear Equations Minimizing Linear Growth Functionals*, vol. 223 of *Progress in Mathematics*. Birkhäuser, Basel, 2004
- [9] J. Astola, P. Haavisto, Y. Neuvo. Vector median filters. *Proc. IEEE*, 78(4):678–689, 1990
- [10] T. L. Austin. An approximation to the point of minimum aggregate distance. *Metron*, 19:10–21, 1959
- [11] Y. Bao, H. Krim. Towards bridging scale-space and multiscale frame analyses. In A. A. Petrosian and F. G. Meyer, editors, *Wavelets in Signal and Image Anal-*

- ysis, vol. 19 of *Computational Imaging and Vision*, chapter 6, Dordrecht, 2001, Kluwer
- [12] L. Bar, B. Berkels, M. Rumpf, G. Sapiro. A Variational Framework for Simultaneous Motion Estimation and Restoration of Motion-Blurred Video. Preprint 2170, Institute for Mathematics and its Applications, Minneapolis, MN, August 2007
- [13] L. Bar, A. Brook, N. Sochen, N. Kiryati. Color image deblurring with impulsive noise. In N. Paragios, O. Faugeras, T. Chan, C. Schnörr, editors, *Variational and Level Set Methods in Computer Vision*, vol. 3752 of *Lecture Notes in Computer Science*, 49–60, Berlin, 2005, Springer
- [14] L. Bar, N. Sochen, N. Kiryati. Variational pairing of image segmentation and blind restoration. In T. Pajdla, J. Matas, editors, *Computer Vision – ECCV 2004*, vol. 3022 of *Lecture Notes in Computer Science*, 166–177, Berlin, 2004, Springer
- [15] L. Bar, N. Sochen, N. Kiryati. Image deblurring in the presence of salt-and-pepper noise. In R. Kimmel, N. Sochen, J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, vol. 3459 of *Lecture Notes in Computer Science*, 107–118, Berlin, 2005, Springer
- [16] L. Bar, N. Sochen, N. Kiryati. Restoration of images with piecewise space-variant blur. In F. Sgallari, A. Murli, N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, vol. 4485 of *Lecture Notes in Computer Science*, 533–544, Berlin, 2007, Springer
- [17] V. Barnett. The ordering of multivariate data. *Journal of the Royal Statistical Society A*, 139(3):318–355, 1976
- [18] M. Barni, F. Buit, F. Bartolini, V. Cappellini. A quasi-Euclidean norm to speed up vector median filtering. *IEEE Trans. Image Processing*, 9(10):1704–1709, 2000
- [19] J. Barral Souto. El modo y otras medias, casos particulares de una misma expresión matemática. *Cuadernos de Trabajo* No. 3, Instituto de Biometria, Universidad Nacional de Buenos Aires, Argentina, 1938
- [20] F. Becker. *Matrix-Valued Filters as Convex Programs*. Diploma thesis, CVGPR group, University of Mannheim, 2004
- [21] G. Bellettini, V. Caselles, M. Novaga. The total variation flow in R^N . *Journal of Differential Equations*, 184(2):475–525, 2002
- [22] W. Benger, H.-C. Hege. Strategies for direct visualization of second-rank tensor fields. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 191–214, Berlin, 2006, Springer

- [23] M. Bertero, P. Boccacci. *Introduction to Inverse Problems in Imaging*. IoP Publishing, Bristol, 1998
- [24] M. Bertero, T. A. Poggio, V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988
- [25] G. Bertrand, A. Imiya, R. Klette, editors. *Digital and Image Geometry*. Volume 2243 of *Lecture Notes in Computer Science*, Springer, Berlin, 2001
- [26] J. Biemond, R. L. Lagendijk, R. M. Mersereau. Iterative methods for image deblurring. *Proc. IEEE*, 78:856–883, 1990
- [27] J. Bigün, G. H. Granlund, J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991
- [28] M. J. Black, P. Anandan. Robust dynamic motion estimation over time. In *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 292–302, Maui, HI, June 1991. IEEE Computer Society Press
- [29] M. J. Black, P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan. 1996
- [30] A. Blake, A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987
- [31] P. Blomgren, T. F. Chan. Color TV: total variation methods for restoration of vector valued images. *IEEE Transactions on Image Processing*, 7(3):304–309, 1998
- [32] P. Blomgren, T. F. Chan, P. Mulet, C. K. Wong. Total variation image restoration: numerical methods and extensions. In *Proc. IEEE International Conference on Image Processing*, vol. 3, 384–387, 1997
- [33] J. P. Boris, D. L. Book. Flux-corrected transport I: SHASTA, a fluid-transport algorithm that works. *Journal of Computational Physics*, 11:38–69, 1973
- [34] S. Boyd, L. Vandenberghe. *Convex Optimization*. 2004, Cambridge University Press
- [35] K. Bredies, D. A. Lorenz, P. Maaß, G. Teschke. A partial differential equation for continuous non-linear shrinkage filtering and its application for analyzing MMG data. In F. Truchetet, editor, *Wavelet Applications in Industrial Processing*, vol. 5266 of *Proceedings of SPIE*, 84–93, Bellingham, 2004, SPIE Press
- [36] M. Breuß. An analysis of discretisations of inverse diffusion equations. *Computing Letters*, 2(3):117–129, 2006

- [37] M. Breuß, T. Brox, A. Bürgel, T. Sonar, J. Weickert. Numerical aspects of TV flow. *Numerical Algorithms*, 41(1):79–101, 2006
- [38] M. Breuß, M. Welk. Staircasing in semidiscrete stabilised inverse diffusion algorithms. *Journal of Computational and Applied Mathematics*, 206(1):520–533, 2007
- [39] R. W. Brockett, P. Maragos. Evolution equations for continuous-scale morphology. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, 125–128, San Francisco, CA, Mar. 1992
- [40] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision – ECCV 2004, Part IV*, vol. 3024 of *Lecture Notes in Computer Science*, 25–36, Berlin, 2004, Springer
- [41] T. Brox, M. Rousson, R. Deriche, J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In N. Petkov, M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, vol. 2756 of *Lecture Notes in Computer Science*, 353–360, Berlin, 2003, Springer
- [42] T. Brox, R. van den Boomgaard, F. Lauze, J. van de Weijer, J. Weickert, P. Mrázek, P. Kornprobst. Adaptive structure tensors and their applications. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 17–47, Berlin, 2006, Springer
- [43] T. Brox, J. Weickert. Nonlinear matrix diffusion for optic flow estimation. In L. Van Gool, editor, *Pattern Recognition*, vol. 2449 of *Lecture Notes in Computer Science*, 446–453, Berlin, 2002, Springer
- [44] T. Brox, M. Welk, G. Steidl, J. Weickert. Equivalence results for TV diffusion and TV regularisation. In L. D. Griffin, M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, vol. 2695 of *Lecture Notes in Computer Science*, 86–100, Berlin, 2003, Springer.
- [45] A. Bruhn. *Variational Optic Flow Computation – Accurate Modelling and Efficient Numerics*. PhD thesis, Saarland University, Saarbrücken, Germany, 2006
- [46] A. Buades, B. Coll, J.-M. Morel. Neighborhood Filters and PDE’s. CMLA Preprint 2005-04, ENS Cachan. *Numerische Mathematik*, 105(1):1–34, 2006.
- [47] B. Burgeth, M. Welk, C. Feddern, J. Weickert. Morphological operations on matrix-valued images. In T. Pajdla, J. Matas, editors, *Computer Vision - ECCV 2004*, vol. 3024 of *Lecture Notes in Computer Science*, 155–167, Berlin, 2004, Springer

- [48] B. Burgeth, M. Welk, C. Feddern, J. Weickert. Mathematical morphology on tensor data using the Loewner ordering. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 357–368, Berlin, 2006, Springer
- [49] V. F. Candela, A. Marquina, S. Serna. A local spectral inversion of a linearized TV model for denoising and deblurring. *IEEE Transactions on Image Processing*, 12:808–816, 2003
- [50] E. Candès, L. Demanet, D. Donoho, L. Ying. Fast discrete curvelet transform. *SIAM Journal on Multiscale Modeling and Simulation*, 5(3):861–899, 2006
- [51] E. J. Candès, D. L. Donoho. Curvelets: A surprisingly effective nonadaptive representation of objects with edges. In A. Cohen, C. Rabut, and L. L. Schumaker, editors, *Curve and Surface Fitting*, Saint-Malo, 2000, Vanderbilt University Press
- [52] E. J. Candès, F. Guo. New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. *Signal Processing*, 82(11):1519–1543, 2002
- [53] V. Caselles, G. Sapiro, D. H. Chung. Vector median filters, inf-sup operations, and coupled PDE’s: Theoretical Connections. *J. Mathematical Imaging and Vision*, 8:109–119, 2000
- [54] F. Catté, P.-L. Lions, J.-M. Morel, T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 32:1895–1909, 1992
- [55] A. Chambolle. Partial differential equations and image processing. *Proc. 1994 IEEE International Conference on Image Processing*, vol. 1, 16–20, Austin, Texas, 1994
- [56] A. Chambolle, R. A. DeVore, N. Lee, B. L. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3):319–335, 1998
- [57] A. Chambolle, B. L. Lucier. Interpreting translationally-invariant wavelet shrinkage as a new image smoothing scale space. *IEEE Transactions on Image Processing*, 10(7):993–1000, 2001
- [58] T. Chan, R. Chan, H. Zhou. A continuation method for total variation denoising problems. In F. T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures, and Implementations*, vol. 2563 of *Proceedings of SPIE*, 314–325, Bellingham, 1995, SPIE Press
- [59] T. F. Chan, C. K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7:370–375, 1998

- [60] T. F. Chan, A. M. Yip, F. E. Park. Simultaneous total variation image inpainting and blind deconvolution. *International Journal of Imaging Systems and Technology*, 15(1):92–102, 2005
- [61] T. F. Chan, H. M. Zhou. Total variation improved wavelet thresholding in image compression. In *Proc. Seventh International Conference on Image Processing*, vol. II, 391–394, Vancouver, Canada, Sept. 2000
- [62] P. Charbonnier, L. Blanc-Féraud, G. Aubert, M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proc. 1994 IEEE International Conference on Image Processing, Austin, TX*, vol. 2, 168–172, 1994
- [63] P. Charbonnier, L. Blanc-Féraud, G. Aubert, M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997
- [64] C. Chef d’Hotel. *Méthodes Géométriques en Vision par Ordinateur et Traitement d’Image: Contributions et Applications*. PhD Thesis, ENS Cachan, France, 2004
- [65] Y. Chen, S. Levine, M. Rao. Variable exponent, linear growth functionals in image restoration. *SIAM Journal on Applied Mathematics*, 66(4):1383–1406, 2006
- [66] A. Cohen, W. Dahmen, I. Daubechies, R. DeVore. Harmonic analysis in the space BV. *Revista Matemática Iberoamericana*, 19:235–262, 2003
- [67] A. Cohen, R. DeVore, P. Petrushev, H. Xu. Nonlinear approximation and the space $BV(\mathbb{R}^2)$. *American Journal of Mathematics*, 121:587–628, 1999
- [68] R. R. Coifman, D. Donoho. Translation invariant denoising. In A. Antoine, G. Oppenheim, editors, *Wavelets in Statistics*, 125–150, New York, 1995, Springer
- [69] R. R. Coifman, A. Sowa. Combining the calculus of variations and wavelets for image enhancement. *Applied and Computational Harmonic Analysis*, 9(1):1–18, 2000
- [70] R. R. Coifman, A. Sowa. New methods of controlled total variation reduction for digital functions. *SIAM Journal on Numerical Analysis*, 39(2):480–498, 2001
- [71] G. B. Coleman, H. C. Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979
- [72] O. Coulon, D. C. Alexander, S. A. Arridge. A regularization scheme for diffusion tensor magnetic resonance images. In M. F. Insana, R. M. Leahy, editors, *Information Processing in Medical Imaging – IPMI 2001*, vol. 2082 of *Lecture Notes in Computer Science*, 92–105, Berlin, 2001, Springer

- [73] F. Dibos, G. Koepfler. Global total variation minimization. *SIAM Journal on Numerical Analysis*, 37(2):646–664, 2000
- [74] M. N. Do, V. Vetterli. Contourlets. In J. Stöckler and G. Welland, editors, *Beyond Wavelets*, 1–27, New York, 2003, Academic Press
- [75] D. L. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995
- [76] D. L. Donoho. Orthonormal ridglets and linear singularities. *SIAM Journal on Mathematical Analysis*, 31(5):1062–1099, 2000
- [77] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994
- [78] E. R. Dougherty, J. Astola, editors. *Nonlinear Filters for Image Processing*. Bellingham, 1999, SPIE Press
- [79] E. C. DuFort, S. P. Frankel. Stability conditions in the numerical treatment of parabolic differential equations. *Mathematical Tables and Other Aids to Computation*, 7:135–152, 1953
- [80] S. Durand, J. Froment. Reconstruction of wavelet coefficients using total-variation minimization. *SIAM Journal on Scientific Computing*, 24(5):1754–1767, 2003
- [81] S. Durand, M. Nikolova. Restoration of wavelet coefficients by minimizing a specially designed objective function. In O. Faugeras, N. Paragios, editors, *Proc. Second IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, Nice, France, Oct. 2003, 145–152, INRIA
- [82] U. Eckhardt: Root images of median filters. *Journal of Mathematical Imaging and Vision*, 19:63–70, 2003
- [83] S. Esedoglu. Stability properties of the Perona–Malik scheme. *SIAM Journal on Numerical Analysis*, 44(3):1297–1313, 2006
- [84] P. Favaro, M. Burger, S. Soatto. Scene and motion reconstruction from defocus and motion-blurred images via anisotropic diffusion. In T. Pajdla, J. Matas, editors, *Computer Vision – ECCV 2004*, vol. 3021 of *Lecture Notes in Computer Science*, 257–269, Berlin, 2004, Springer
- [85] C. Feddern, J. Weickert, B. Burgeth. Level-set methods for tensor-valued images. In O. Faugeras, N. Paragios, editors, *Proc. Second IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, Nice, France, Oct. 2003, 65–72, INRIA

- [86] C. Feddern, J. Weickert, B. Burgeth, M. Welk. Curvature-driven PDE methods for matrix-valued images. *International Journal of Computer Vision*, 69(1):91–103, 2006
- [87] S. P. Fekete, J. S. B. Mitchell, K. Beurer. On the continuous Fermat–Weber problem. *Operations Research*, 53(1):61–76, 2005
- [88] X. Feng, A. Prohl. Analysis of total variation flow and its finite element approximations. *ESAIM: Mathematical Models and Methods in the Applied Sciences*, 37(3):533–556, 2002
- [89] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, W. T. Freeman. Removing camera shake from a single image. *ACM Transactions on Graphics*, 25(3):787–794, 2006
- [90] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Kluwer, Dordrecht, 1988
- [91] W. Förstner, E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, 281–305, Interlaken, Switzerland, June 1987
- [92] G. Gerig, O. Kübler, R. Kikinis, F. A. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992
- [93] S. Geršgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izvestiya Akademii Nauk SSSR, otdelenie matematičeskikh i estestvennyh nauk*, 1, 749–754, 1931 (in German)
- [94] G. Gilboa, N. A. Sochen, Y. Y. Zeevi. Regularized shock filters and complex diffusion. In A. Heyden, G. Sparr, M. Nielsen, P. Johansen, editors, *Computer Vision – ECCV 2002*, vol. 2350 of *Lecture Notes in Computer Science*, 399–413, Berlin, 2002, Springer
- [95] G. Gilboa, N. Sochen, Y. Y. Zeevi. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Transactions on Image Processing*, 11(7):689–703, 2002
- [96] G. Gilboa, N. Sochen, Y. Y. Zeevi. Image sharpening by flows based on triple well potentials. *Journal of Mathematical Imaging and Vision*, 20:121–131, 2004
- [97] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer, Dordrecht, 1995
- [98] L. D. Griffin. Mean, median and mode filtering of images. *Proceedings of the Royal Society of London, Series A*, 456(2004):2995–3004, 2000

- [99] F. Guichard, J.-M. Morel. Partial differential equations and image iterative filtering. In: I. S. Duff, G. A. Watson, editors, *The State of the Art in Numerical Analysis*, vol. 63 of *IMA Conference Series (New Series)*, 525–562, Oxford, 1997, Clarendon Press
- [100] F. Guichard, J.-M. Morel. A note on two classical shock filters and their asymptotics. In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision*, vol. 2106 of *Lecture Notes in Computer Science*, 75–84, Berlin, 2001, Springer
- [101] K. Hahn, S. Prigarin, S. Heim, K. Hasan. Random noise in diffusion tensor imaging, its destructive impact and some corrections. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 107–119, Berlin, 2006, Springer
- [102] K. Hahn, S. Prigarin, B. Pütz. Edge preserving regularization and tracking for diffusion tensor imaging. In W. J. Niessen, M. A. Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001*, vol. 2208 of *Lecture Notes in Computer Science*, 195–203, Berlin, 2001, Springer
- [103] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. MIT Press, Cambridge, Massachusetts, 1986
- [104] M. Hanke, J. G. Nagy. Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques. *Inverse Problems*, 12:157–173, 1996
- [105] M. Hanke, J. G. Nagy, C. Vogel. Quasi-Newton approach to nonnegative image restoration. *Linear Algebra and Applications*, 316:223–236, 2000
- [106] S. Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. Academic Press, New York, 1978
- [107] W. Hinterberger, O. Scherzer, C. Schnörr, J. Weickert. Analysis of optical flow models in the framework of calculus of variations. *Numerical Functional Analysis and Optimization*, 23(1/2):69–89, 2002
- [108] P. J. Huber. *Robust Statistics*. Wiley, New York, 1981
- [109] R. A. Hummel, B. Kimia, S. W. Zucker. Deblurring Gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38:66–80, 1987
- [110] T. Iijima. Basic theory of pattern observation. In *Papers of Technical Group on Automata and Automatic Control*. IECE, Japan, Dec. 1959 (in Japanese)
- [111] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, 123–134, Nov. 1963 (in English)

- [112] B. Jawerth, P. Lin, E. Sinzinger. Lattice Boltzmann models for anisotropic diffusion of images. *Journal of Mathematical Imaging and Vision*, 11:231–237, 1999
- [113] J. Kamm, J. G. Nagy. Kronecker product and SVD approximations for separable spatially variant blurs. In F. T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures, and Implementations*, vol. 3461 of *Proceedings of SPIE*, 358–369, Bellingham, 1998, SPIE Press
- [114] J. Kamm, J. G. Nagy. Kronecker product and SVD approximations in image restoration. *Linear Algebra and Applications*, 284:177–192, 1998
- [115] J. Kamm, J. G. Nagy. Optimal Kronecker product approximation of block Toeplitz matrices. *SIAM Journal on Matrix Analysis and Applications*, 22:155–172, 2000
- [116] S. L. Keeling, R. Stollberger. Nonlinear anisotropic diffusion filters for wide range edge sharpening. *Inverse Problems*, 18:175–190, 2002
- [117] J. Keuchel, C. Schnörr, C. Schellewald, D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11):1364–1379, 2003
- [118] B. B. Kimia, A. Tannenbaum, S. W. Zucker. On the evolution of curves via a function of curvature. I. The classical case. *Journal of Mathematical Analysis and its Applications*, 163:438–458, 1992
- [119] B. B. Kimia, S. W. Zucker. Analytic inverse of discrete Gaussian blur. *Optical Engineering*, 32:166–176, 1993
- [120] R. Kimmel. *Numerical Geometry of Images: Theory, Algorithms and Applications*. Springer, New York, 2003
- [121] N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3):234–253, 2001
- [122] R. Klette, R. Zamperoni. *Handbook of Image Processing Operators*. Wiley, New York, 1996
- [123] P. Kornprobst, R. Deriche, G. Aubert. Nonlinear operators in image restoration. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 325–330, San Juan, Puerto Rico, June 1997, IEEE Computer Society Press
- [124] A. Koschan, M. Abidi. A comparison of median filter techniques for noise removal in color images. In *Proc. Seventh German Workshop on Color Image Processing*, Erlangen, Germany, 69–79, 2001

- [125] H. P. Kramer, J. B. Bruckner. Iterations of a non-linear transformation for enhancement of digital images. *Pattern Recognition*, 7:53–58, 1975
- [126] H. W. Kuhn. A note on Fermat’s Problem. *Mathematical Programming*, 4(1):98–107, 1973
- [127] D. Kuzmin, R. Löhner, S. Turek, editors. *Flux-Corrected Transport*. Springer, Berlin, 2005
- [128] D. Labate, W. Lim, G. Kutyniok, G. Weiss. Sparse multidimensional representation using shearlets. In M. Papadakis, A. F. Laine, M. A. Unser, editors, *Wavelets XI*, vol. 5914 of *Proceedings of SPIE*, 254–262, Bellingham, 2005, SPIE Press
- [129] R. L. Lagendijk, J. Biemond, D. E. Boeke. Regularized iterative image restoration with ringing reduction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36:1874–1888, 1988
- [130] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002
- [131] S. Levine, Y. Chen, J. Stanich. Image Restoration via Nonstandard Diffusion. Technical Report 04-01, Dept. of Mathematics and Computer Science, Duquesne University, Pittsburgh, 2004
- [132] H.-Y. Lin, C.-H. Chang. Depth from motion and defocus blur. *Optical Engineering*, 45(12):127201, 1–12, 2006
- [133] F. Liu, X. E. Ruan. Wavelet-based diffusion approaches for signal denoising. *Signal Processing*, 87:1138–1146, 2007
- [134] T. Lu, P. Neittaanmäki, X.-C. Tai. A parallel splitting up method and its application to Navier–Stokes equations. *Applied Mathematics Letters*, 4(2):25–29, 1991
- [135] B. A. Mair, Z. Réti, D. C. Wilson, E. A. Geiser, B. David. A q -series approach to deblurring the discrete Gaussian. *Computer Vision and Image Understanding*, 66:247–254, 1997
- [136] F. Malgouyres. Combining total variation and wavelet packet approaches for image deblurring. In *Proc. First IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 57–64, Vancouver, Canada, July 2001, IEEE Computer Society Press
- [137] F. Malgouyres. Mathematical analysis of a model which combines total variation and wavelet for image restoration. *Informacionnĭe Processĭ*, 2(1):1–10, 2002

- [138] A. Marquina, S. Osher. A new time dependent model based on level set motion for nonlinear deblurring and noise removal. In M. Nielsen, P. Johansen, O. F. Olsen, J. Weickert, editors, *Scale-Space Theories in Computer Vision*, vol. 1682 of *Lecture Notes in Computer Science*, 429–434, Berlin, 1999, Springer
- [139] G. Medioni, M. S. Lee, C. K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, Amsterdam, 2000
- [140] N. Megiddo. The weighted Euclidean 1-center problem. *Mathematics of Operations Research*, 8(4):498–504, 1983
- [141] E. Mémin, P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002
- [142] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. vol. 22 of *University Lecture Series*, AMS, Providence, 2001
- [143] M. Moakher, P. Batchelor. Symmetric positive-definite matrices: from geometry to applications and visualization. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 285–298, Berlin, 2006, Springer
- [144] P. Mrázek, J. Weickert. Rotationally invariant wavelet shrinkage. In B. Michaelis, G. Krell, editors, *Pattern Recognition*, vol. 2781 of *Lecture Notes in Computer Science*, 156–163, Berlin, 2003, Springer
- [145] P. Mrázek, J. Weickert, G. Steidl. Diffusion-inspired shrinkage functions and stability results for wavelet denoising. *International Journal of Computer Vision*, 64(2/3):171–186, 2005
- [146] P. Mrázek, J. Weickert, G. Steidl, M. Welk. On iterations and scales of nonlinear filters. In O. Drbohlav, editor, *Proc. Eighth Computer Vision Winter Workshop*, 61–66, Valtice, Czech Republic, Feb. 2003, Czech Pattern Recognition Society
- [147] D. Mumford, J. Shah. Boundary detection by minimizing functionals, I. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA*, 22–26, 1985
- [148] J. Nagy, M. Ng, L. Perrone. Kronecker product approximations for image restoration with reflexive boundary conditions. *SIAM Journal on Matrix Analysis and Applications*, 25(3):829–841, 2003
- [149] J. G. Nagy, D. P. O’Leary. Fast iterative image restoration with a spatially varying PSF. In F. T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures, and Implementations*, vol. 3162 of *Proceedings of SPIE*, 388–399, Bellingham, 1997, SPIE Press

- [150] J. G. Nagy, D. P. O’Leary. Restoring images degraded by spatially-variant blur. *SIAM Journal on Scientific Computing*, 19:1063–1082, 1998
- [151] J. G. Nagy, Z. Strakoš. Enforcing nonnegativity in image reconstruction algorithms. In D. C. Wilson et al., editors, *Mathematical Modeling, Estimation, and Imaging*, vol. 4121 of *Proc. SPIE*, 182–190, 2000
- [152] M. Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002
- [153] M. Nikolova. Weakly constrained minimization: Application to the estimation of images and signals involving constant regions. *Journal of Mathematical Imaging and Vision*, 21:155–175, 2004
- [154] N. Nordström. Biased anisotropic diffusion – a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8:318–327, 1990
- [155] S. Osher, L. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27:919–940, 1990
- [156] S. Osher, L. Rudin. Shocks and other nonlinear filtering applied to image processing. In A. G. Tescher, editor, *Applications of Digital Image Processing XIV*, vol. 1567 of *Proceedings of SPIE*, 414–431, Bellingham, 1991, SPIE Press
- [157] S. Osher, L. Rudin. Total variation based image restoration with free local constraints. *Proceedings of the IEEE ICIP*, 31–35, Austin, Texas, 1994
- [158] S. Osher, J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988
- [159] G. J. M. Parker, J. A. Schnabel, M. R. Symms, D. J. Werring, G. J. Barker. Non-linear smoothing for reduction of systematic and random errors in diffusion tensor imaging. *Journal of Magnetic Resonance Imaging*, 11:702–710, 2000
- [160] X. Pennec, P. Fillard, N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006
- [161] P. Perona, J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990
- [162] A. Petrovic, O. Divorra Escoda, P. Vandergheynst. Multiresolution segmentation of natural images: From linear to non-linear scale-space representations. *IEEE Transactions on Image Processing*, 13(8):1104–1114, 2004

- [163] C. Pierpaoli, P. Jezzard, P. J. Basser, A. Barnett, G. Di Chiro. Diffusion tensor MR imaging of the human brain. *Radiology*, 201(3):637–648, 1996
- [164] I. Pollak, A. S. Willsky, H. Krim. Scale space analysis by stabilized inverse diffusion equations. In B. ter Haar Romeny, L. Florack, J. Koenderink, M. Viergever, editors, *Scale-Space Theory in Computer Vision*, vol. 1252 of *Lecture Notes in Computer Science*, 200–211, Berlin, 1997, Springer
- [165] I. Pollak, A. S. Willsky, H. Krim. Image segmentation and edge enhancement with stabilized inverse diffusion equations. *IEEE Transactions on Image Processing*, 9(2):256–266, 2000
- [166] C. Poupon, J. Mangin, V. Frouin, J. Régis, F. Poupon, M. Pachot-Clouard, D. Le Bihan, I. Bloch. Regularization of MR diffusion tensor maps for tracking brain white matter bundles. In W. M. Wells, A. Colchester, S. Delp, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 1998*, vol. 1496 of *Lecture Notes in Computer Science*, 489–498, Berlin, 1998, Springer
- [167] T. Preußner, M. Rumpf. An adaptive finite element method for large scale image processing. *Journal of Visual Communication and Image Representation*, 11(2):183–195, 2000
- [168] C. B. Price, P. Wambacq, A. Oosterlinck. Computing with reaction-diffusion systems: applications in image processing. In D. Roose et al., editors, *Continuation and Bifurcations: Numerical Techniques and Applications*, 379–387, Dordrecht, 1990, Kluwer
- [169] A. R. Rao, B. G. Schunck. Computing oriented texture fields. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 53:157–185, 1991
- [170] L. Remaki, M. Cheriet. Numerical schemes of shock filter models for image enhancement and restoration. *Journal of Mathematical Imaging and Vision*, 18(2):153–160, 2003
- [171] J. L. Richardson, R. C. Ferrell, L. N. Long. Unconditionally stable explicit algorithms for nonlinear fluid dynamics problems. *Journal of Computational Physics*, 104:69–74, 1993
- [172] W. Ring. Structural properties of solutions to total variation regularization problems. *Mathematical Modelling and Numerical Analysis*, 34:799–810, 2000
- [173] M. Rousson, T. Brox, R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. Technical report no. 4695, Odysée, INRIA Sophia-Antipolis, France, 2003
- [174] L. I. Rudin, S. Osher, E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992

- [175] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001
- [176] G. Sapiro, D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 5(11):1582–1586, 1996
- [177] A. E. Savakis, R. L. Easton jr. Blur identification based on higher order spectral nulls. In T. J. Schulz, D. L. Snyder, *Image Reconstruction and Restoration*, vol. 2302 of *Proceedings of SPIE*, 168–177, Bellingham, 1994, SPIE Press
- [178] A. E. Savakis, H. J. Trussell. Blur identification by residual spectral matching. *IEEE Transactions on Image Processing*, 2(2):141–151, 1993
- [179] A. E. Savakis, H. J. Trussell. On the accuracy of PSF representation in image restoration. *IEEE Transactions on Image Processing*, 2(2):252–259, 1993
- [180] J. G. M. Schavemaker, M. J. T. Reinders, J. J. Gerbrands, E. Backer. Image sharpening by morphological filtering. *Pattern Recognition*, 33:997–1012, 2000
- [181] C. Schnörr. Unique reconstruction of piecewise smooth images by minimizing strictly convex non-quadratic functionals. *Journal of Mathematical Imaging and Vision*, 4:189–198, 1994
- [182] C. Schnörr. Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Proc. Twelfth International Conference on Pattern Recognition*, vol. A, 661–663, Jerusalem, Israel, Oct. 1994, IEEE Computer Society Press
- [183] D. R. Seymour. Note on Austin’s “An approximation to the point of minimum aggregate distance”. *Metron*, 28:412–421, 1970
- [184] J. Shen. A note on wavelets and diffusion. *Journal of Computational Analysis and Applications*, 5(1):147–159, 2003
- [185] J. Shen, G. Strang. On wavelet fundamental solutions to the heat equation – heatlets. *Journal of Differential Equations*, 161(2):403–421, 2000
- [186] A. S. Sherstinsky, R. W. Picard. M-Lattice: from morphogenesis to image processing. *IEEE Transactions on Image Processing*, 5(7):1137–1150, 1996
- [187] C. L. Siegel. *Symplectic Geometry*. Academic Press, New York, 1964
- [188] E. P. Simoncelli. Design of multidimensional derivative filters. In *Proc. 1994 IEEE International Conference on Image Processing*, vol. 1, 790–793, Austin, Texas, Nov. 1994

- [189] N. Sochen, R. Kimmel, F. Bruckstein. Diffusions and confusions in signal and image processing. *Journal of Mathematical Imaging and Vision*, 14(3):195–210, 2001
- [190] M. Šorel, J. Flusser. Restoration of color images degraded by space-variant motion blur. In W. G. Kropatsch, M. Kampel, A. Hanbury, editors, *Computer Analysis of Images and Patterns*, vol. 4673 of *Lecture Notes in Computer Science*, 450–457, Berlin, 2007, Springer
- [191] G. Steidl, J. Weickert, T. Brox, P. Mrázek, M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686–713, 2004
- [192] B. M. ter Haar Romeny, L. M. J. Florack, M. de Swart, J. Wilting, M. A. Viergever. Deblurring Gaussian blur. In F. L. Bookstein, J. S. Duncan, N. Lange, D. C. Wilson, editors, *Mathematical methods in medical imaging III*, vol. 2299 of *Proceedings of SPIE*, 139–148, Bellingham, 1994, SPIE Press
- [193] D. Theis. *Dekonvolution digitaler Bilder*. Diploma thesis, Saarland University, Saarbrücken, 2004 (in German)
- [194] P. L. Torroba, N. L. Cap, H. J. Rabal, W. D. Furlan. Fractional order mean in image processing. *Optical Engineering*, 33(2):528–534, 1994
- [195] D. Tschumperlé, R. Deriche. Diffusion tensor regularization with constraints preservation. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Kauai, HI, Dec. 2001, 948–953, IEEE Computer Society Press
- [196] D. Tschumperlé, R. Deriche. Diffusion PDE's on vector-valued images. *IEEE Signal Processing Magazine*, 19(5):16–25, 2002
- [197] V. I. Tsurkov. An analytical model of edge protection under noise suppression by anisotropic diffusion. *Journal of Computer and Systems Sciences International*, 39(3):437–440, 2000
- [198] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Menlo Park, 1971
- [199] R. van den Boomgaard, J. van de Weijer. Least squares and robust estimation of local image structure. In L. D. Griffin and M. Lillholm, editors, *Scale Space Methods in Computer Vision*, vol. 2695 of *Lecture Notes in Computer Science*, 237–254, Berlin, 2003, Springer
- [200] A. Vilanova, S. Zhang, G. Kindlmann, D. Laidlaw. An introduction to visualization of diffusion tensor imaging and its applications. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 121–153, Berlin, 2006, Springer

- [201] R. Vio, J. Nagy, L. Tenorio, P. Andreani, C. Baccigalupi, W. Wamsteker. Digital deblurring of CMB maps I: performance and efficient implementation. *Astronomy and Astrophysics*, 401:389–404, 2003
- [202] R. Vio, J. Nagy, L. Tenorio, P. Andreani, C. Baccigalupi, W. Wamsteker. Digital deblurring of CMB maps II: asymmetric point spread function. *Astronomy and Astrophysics*, 401:835–843, 2003
- [203] W. Wang. On the design of optimal derivative filters for coherence-enhancing diffusion filtering. In *Proc. 2004 International Conference on Computer Graphics, Imaging and Visualization*, 35–40, Penang, Malaysia, July 2004, IEEE Computer Society Press
- [204] J. Weickert. Scale-space properties of nonlinear diffusion filtering with a diffusion tensor. Technical Report 110, Laboratory of Technomathematics, University of Kaiserslautern, Germany, Oct. 1994
- [205] J. Weickert. A review of nonlinear diffusion filtering. In B. ter Haar Romeny, L. Florack, J. Koenderink, M. Viergever, editors, *Scale-Space Theory in Computer Vision*, vol. 1252 of *Lecture Notes in Computer Science*, 3–28, Berlin, 1997, Springer
- [206] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998
- [207] J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2/3):111–127, 1999
- [208] J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3–4):199–210, 1999
- [209] J. Weickert. Coherence-enhancing shock filters. In B. Michaelis, G. Krell, editors, *Pattern Recognition*, vol. 2781 of *Lecture Notes in Computer Science*, 1–8, Berlin, 2003, Springer
- [210] J. Weickert, B. Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox. In F. Solina, W. G. Kropatsch, R. Klette, R. Baicsy, editors, *Advances in Computer Vision*, 1–10, Wien, 1997, Springer
- [211] J. Weickert, T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, vol. 313 of *Contemporary Mathematics*, 251–268, Providence, 2002, AMS
- [212] J. Weickert, C. Feddern, M. Welk, B. Burgeth, T. Brox. PDEs for tensor image processing. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 399–414, Berlin, 2006, Springer

- [213] J. Weickert, H. Hagen, editors. *Visualization and Processing of Tensor Fields*. Springer, Berlin, 2006
- [214] J. Weickert, H. Scharr. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, 13(1/2):103–118, 2002
- [215] J. Weickert, B. M. ter Haar Romeny, M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, Mar. 1998
- [216] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tôhoku Mathematics Journal*, 43:355–386, 1937
- [217] M. Welk, F. Becker, C. Schnörr, J. Weickert. Matrix-valued filters as convex programs. In R. Kimmel, N. Sochen, J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, vol. 3459 of *Lecture Notes in Computer Science*, 204–216, Berlin, 2005, Springer
- [218] M. Welk, C. Feddern, B. Burgeth, J. Weickert. Median filtering of tensor-valued images. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, vol. 2781 of *Lecture Notes in Computer Science*, 17–24, Berlin, 2003, Springer
- [219] M. Welk, C. Feddern, B. Burgeth, J. Weickert. Tensor median filtering and M-smoothing. In J. Weickert, H. Hagen, editors, *Visualization and Processing of Tensor Fields*, 345–356, Berlin, 2006, Springer
- [220] M. Welk, J. G. Nagy. Variational deconvolution of multi-channel images with inequality constraints. In J. Martí, J. M. Benedí, A. M. Mendonça, J. Serrat, editors, *Pattern Recognition and Image Analysis*, vol. 4477 of *Lecture Notes in Computer Science*, 386–393, Berlin, 2007, Springer
- [221] M. Welk, G. Steidl, J. Weickert. Locally analytic schemes: a link between diffusion filtering and wavelet shrinkage. *Applied and Computational Harmonic Analysis*, 24:195–224, 2008
- [222] M. Welk, D. Theis, T. Brox, J. Weickert. PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. In R. Kimmel, N. Sochen, J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, vol. 3459 of *Lecture Notes in Computer Science*, 585–597, Berlin, 2005, Springer
- [223] M. Welk, D. Theis, J. Weickert. Variational deblurring of images with uncertain and spatially variant blurs. In W. Kropatsch, R. Sablatnig, A. Hanbury, editors, *Pattern Recognition*, vol. 3663 of *Lecture Notes in Computer Science*, 485–492, Berlin, 2005, Springer

- [224] M. Welk, J. Weickert. Semidiscrete and discrete well-posedness of shock filtering. In C. Ronse, L. Najman, E. Decencière, editors, *Mathematical Morphology: 40 Years On*. vol. 30 of *Computational Imaging and Vision*, 311–320, Dordrecht, 2005, Springer
- [225] M. Welk, J. Weickert, F. Becker, C. Schnörr, C. Feddern, B. Burgeth. Median and related local filters for tensor-valued images. *Signal Processing*, 87:291–308, 2007
- [226] M. Welk, J. Weickert, I. Galić. Theoretical foundations for spatially discrete 1-D shock filtering. *Image and Vision Computing*, 25(4):455–463, 2007
- [227] M. Welk, J. Weickert, G. Steidl. A four-pixel scheme for singular differential equations. In R. Kimmel, N. Sochen, J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, vol. 3459 of *Lecture Notes in Computer Science*, 585–597, Berlin, 2005, Springer
- [228] M. Welk, J. Weickert, G. Steidl. From tensor-driven diffusion to anisotropic wavelet shrinkage. In H. Bischof, A. Leonardis, A. Pinz, editors, *Computer Vision – ECCV 2006*, vol. 3951 of *Lecture Notes in Computer Science*, 391–403, Berlin, 2006, Springer
- [229] C. Westin, S. E. Maier, B. Khidhir, P. Everett, F. A. Jolesz, R. Kikinis. Image processing for diffusion tensor magnetic resonance imaging. In C. Taylor, A. Colchester, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 1999*, vol. 1679 of *Lecture Notes in Computer Science*, 441–452, Berlin, 1999, Springer
- [230] R. T. Whitaker, S. M. Pizer. A multi-scale approach to nonuniform diffusion. *Computer Vision, Graphics and Image Processing: Image Understanding*, 57:99–110, 1993
- [231] N. Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. Cambridge (Mass.), 1949, The MIT Press
- [232] G. Winkler, V. Aurich, K. Hahn, A. Martin, K. Rodenacker. Noise reduction in images: some recent edge-preserving methods. *Pattern Recognition and Image Analysis*, 9(4):749–766, 1999
- [233] Y.-L. You, M. Kaveh. A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing*, 5(3):416–428, 1996
- [234] Y.-L. You, M. Kaveh. Anisotropic blind image restoration. In *Proc. 1996 IEEE International Conference on Image Processing*, vol. 2, 461–464, Lausanne, Switzerland, Sept. 1996

- [235] M. E. Zervakis, A. K. Katsaggelos, T. M. Kwon. A class of robust entropic functionals for image restoration. *IEEE Transactions on Image Processing*, 4(6):752–773, 1995
- [236] L. Zhukov, K. Munseth, D. Breen, R. Whitaker, A. H. Barr. Level set modeling and segmentation of DT-MRI brain data. *Journal of Electronic Imaging*, 12(1):125–133, 2003

References added in revision:

- [237] M. Welk, A. Kleefeld and M. Breuß (2016): Quantile filtering of colour images via symmetric matrices. *Mathematical Morphology: Theory and Applications*, accepted 2016, in press.